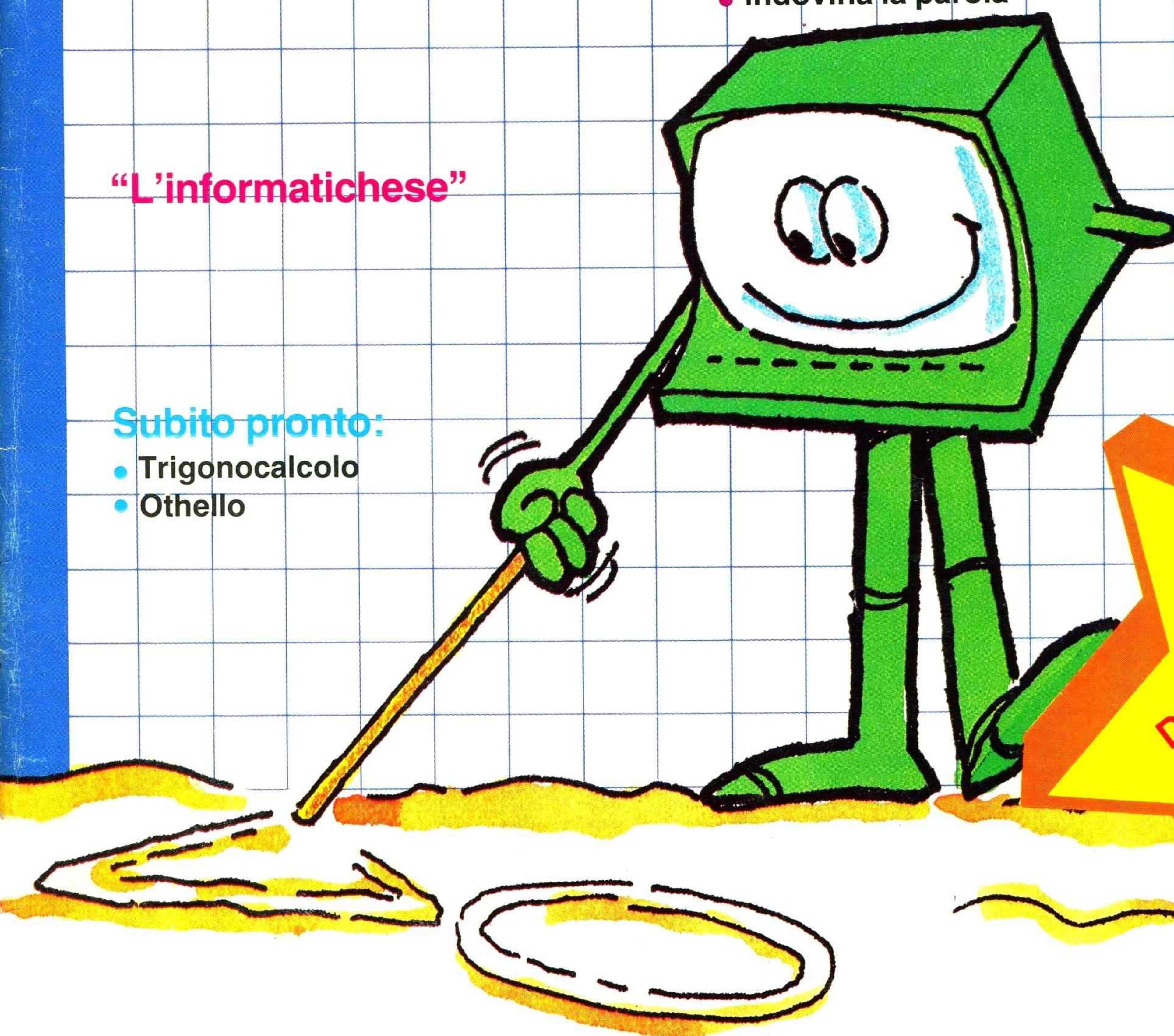
ABC
PERSONAL COMPUTER
QUIZ
GRUPPO
EDITORIALE
JACKSON
1
Spedizione
in abbonamento
Postale
Gruppo II/70
L. 3.000
Esercizi pratici in BASIC
Programmi ★ Problemi ★ Soluzioni
Dalle istruzioni al programma:
• PRINT, INPUT
DATA, READ, RESTORE
• coniugazione dei verbi
• indovina la parola
A domanda ... risponde
Softmarket:
Word processor
"L'informatichese"
Subito pronto:
• Trigonocalcolo
• Othello
DALLA TEORIA
ALLA
PRATICA
DEL BASIC

Caro lettore,

mi permetta di essere curioso e di domandarle perchè, tra tante pubblicazioni di informatica, ha acquistato proprio ABC ... QUIZ!
Forse perchè è firmata dalla Jackson?
O perchè ha appena terminato il corso di Basic di ABC e ne è rimasto entusiasta?
O, ancora, perchè era impaziente di passare dalla teoria dei libri alla pratica del calcolatore?
Quale che sia la sua risposta voglio anticiparle che non ha gettato il suo denaro!
La Jackson, infatti, è garanzia di serietà e di specializzazione.
ABC Personal Computer è stato giudicato uno dei migliori Corsi di Basic mai pubblicati.
ABC...QUIZ è nato proprio per fornire esercizi pratici, programmi problemi e soluzioni a chi conosce il Basic e vuole applicarlo.
Strutturato in 10 fascicoli da rilegare in volume, ABC...Quiz descrive il problema e la sua soluzione; spiega il tracciamento dei diagrammi a blocchi; presenta i listati dei diversi programmi; segnala le possibili varianti a seconda dei diversi dialetti Basic; commenta i programmi professionali di largo uso: spreadsheet, word processor, data base.
In più, ABC...Quiz, per saggiare la sua abilità, le farà molte domande di cui troverà la risposta nel fascicolo successivo e, in ogni numero, le "regalerà" programmi di gioco o di utilità.
Come vede, ABC...Quiz è davvero la pubblicazione che si aspettava e che solo la Jackson poteva realizzare.
Ciò non toglie che debba ringraziarla della sua scelta e assicurarla che, per tutti questi 10 numeri, non deluderò le sue aspettative.

Il Direttore Editoriale
Roberto Pancaldi

ABC...QUIZ
Pubblicazione a fascicoli quindicinali edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore Editoriale: Roberto Pancaldi
Capo Redattore:
Enrico Odetti
Comitato di Redazione:
Paolo Capobussi, Piero Dell'Orco, Gianni Giaccaglini, Daria Gianni, Sergio Mello-Grand, Riccardo Paolillo Carlo Tognoni
Segretaria di Redazione:
Marta Menegardo
Grafica e impaginazione:
Studio Nuovidea - Via Longhi, 16 - Milano
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Reweba
Via Volta, 177 - Brescia
Direzione e Redazione:
Via Rosellini, 12 - Milano (20124)
Tel. 02/6880951/5

Numero in attesa dell'autorizzazione del tribunale di Milano.
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 3.000.
Abbonamento L. 30.000.
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12 20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati saranno disponibili per un anno dal completamento dell'opera e potranno essere prenotati presso le edicole o richiesti direttamente alla casa editrice.
Non vengono effettuate spedizioni contrassegno.

Gruppo Editoriale Jackson
Direzione Editoriale
Giampietro Zanga
e Paolo Reina

SOMMARIO

SUL PROSSIMO FASCICOLO:

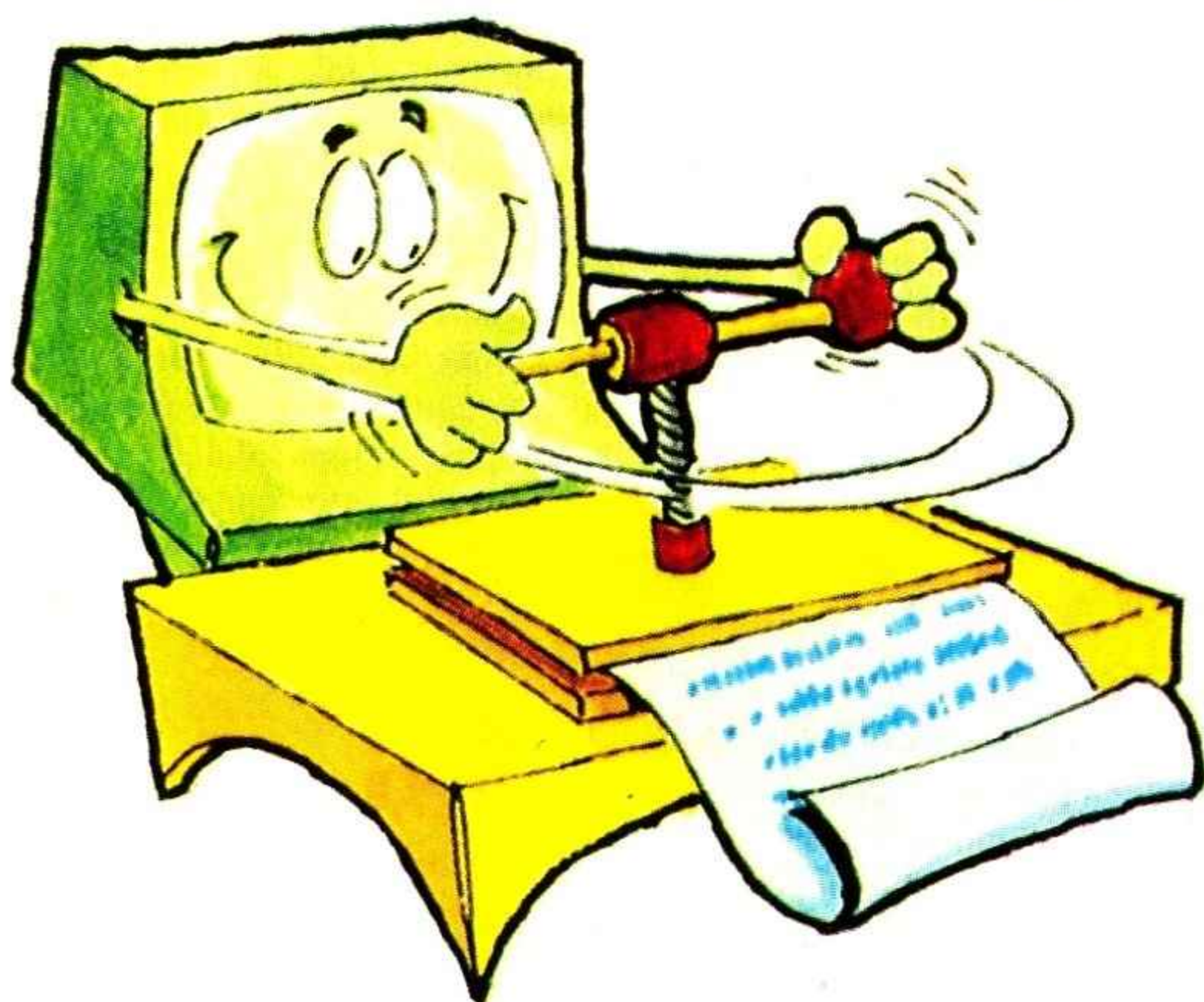
I file sequenziali: scrittura e lettura

Istruzioni:

OPEN CLOSE
PRINT # INPUT #
LINE INPUT # GET #
INPUT$

WordStar

Forse non tutto si può studiare in "autoistruzione" o da autodidatti, ma fra le discipline che più si prestano a questo metodo di apprendimento brilla l'Informatica, o almeno quella parte dell'Informatica che si occupa dell'utilizzo del computer.
Questo per almeno due motivi: da una parte perché la logica rigorosa e stringente che occorre rispettare per impostare la soluzione dei problemi sul calcolatore ha un peso dominante sulla fantasia e creatività - pure necessarie per fare un buon analista/programmatore - ma soprattutto perché per conoscere una macchina la cosa migliore è usarla, e il calcolatore (il Personal) è ormai uno strumento alla portata di tutti, non solo per il prezzo, ma anche per la facilità d'uso.
E allora continuiamo a imparare da soli, passando ora dalla teoria alla pratica. Questa pubblicazione comincia dove finisce ABC Personal Computer, ma più in generale si propone a chiunque abbia acquisito le conoscenze di base sull'hardware e sul software di un computer, una certa confidenza con il set delle istruzioni BASIC e i principi fondamentali da osservare nella risoluzione di un problema con il calcolatore.
La struttura dell'opera è agile e articolata: ci sono domande a cui dovete provare a rispondere da soli e di cui troverete la risposta successivamente.
Ci sono problemi proposti con la soluzione completa e dettagliata, da studiare con attenzione proprio per imparare; c'è il softmarket con la presentazione dei pacchetti di software (programmi professionali) più noti e più usati sui personal computer. C'è un po' di storia e attualità e, "dulcis in fundo", dei programmi pronti, spesso dei giochi, di cui spieghiamo le istruzioni più caratteristiche.
Ogni capitolo di ABC QUIZ è dedicato ad alcune importanti istruzioni BASIC.
Di queste diamo dapprima le regole d'uso (la sintassi) e poi mostriamo come impiegarle in programmi applicativi.
Il nostro obiettivo è di aiutarvi a scrivere da voi stessi i vostri programmi e a trovare nella programmazione il piacere sottile nel vedere il calcolatore fare proprio quello che volete.
Programmare è un'arte!
Per imparare a programmare non bisogna fare altro che programmare.
Nel labirinto dei dialetti BASIC abbiamo cercato di usare quelle istruzioni che sono più ampiamente note. Ciò non toglie che quando presentiamo un programma, che giri effettivamente su una macchina, ne diamo la specifica versione di BASIC.
E adesso via ... auguri e buon lavoro.

dalle istruzioni al programma

Qualunque programma contiene sempre delle istruzioni per l'ingresso e l'uscita di dati e messaggi: PRINT e INPUT sono le più importanti. Il BASIC, però, usa anche altre istruzioni che permettono d'inserire direttamente i dati nel programma: DATA, READ e RESTORE.
Queste istruzioni possono sembrare le più banali e facili, e non richiedere doti particolari al programmatore. Niente di più pericoloso. In realtà un buon uso di queste istruzioni migliora la qualità del programma, lo rende efficiente ed evita tante complicazioni al momento del suo utilizzo. Pensiamo fondamentale ricordare dapprima quali sono le regole sintattiche per il loro uso anche a costo di annoiare chi già le conosce.
Subito dopo le vedremo all'opera in programmi concreti.
Non si tratta di programmi professionali, ma a carattere didattico: come tali il loro obiettivo è quello d'insegnare i concetti e fornire strumenti di lavoro.

PRINT

Invia sullo schermo il contenuto delle variabili indicate, ma può anche visualizzare direttamente delle espressioni. PRINT si comporta nello stesso modo in tutti i dialetti BASIC. Le varianti di PRINT sono nell'uso delle virgole o dei punti e virgola come separatori. Una virgola, o un punto e virgola, dopo l'ultima variabile o espressione indicata, impedisce che il cursore vada a capo per le stampe successive. Per esempio:

```
100 PRINT "CIAO";
110 PRINT "  MI CHIAMO ABC"

RUN

CIAO MI CHIAMO ABC
```

Notate lo spazio lasciato davanti a MI per staccare da CIAO.
L'uso del separatore punto e virgola nella PRINT permette di trasformare questa istruzione d'uscita in un "concatenatore" di stringhe. Facciamo un altro esempio:

```
100 A$ = "PRECIPITE"
110 B$ = "VOLISSIMEVOL"
120 C$ = "MENTE"
130 PRINT A$; B$; C$

PRECIPITEVOLISSIMEVOLMENTE
```

Infatti poiché non viene lasciato alcun spazio tra una stringa stampata e la successiva, ciò equivale a stampare una lunga stringa intera.
Se in una PRINT si usa la funzione CHR$ si possono anche stampare quei caratteri difficili come le virgolette (o RETURN ...). Per esempio se vogliamo stampare la frase:

```
"Input" è sinonimo d'ingresso di dati
```

Possiamo usare una PRINT in questo modo:

```
PRINT CHR$ (34); "Input"; CHR$ (34); "è
sinonimo d'ingresso di dati"
```

Il numero 34 indicato nelle due CHR$ è il codice ASCII delle virgolette ("). Nello stesso modo si possono stampare altri caratteri particolari tra i quali:

CHR$ (7)	Campanello
CHR$ (13)	Ritorno a capo
CHR$ (32)	Spazio
CHR$ (34)	Virgolette
CHR$ (44)	Virgola

Una sola PRINT da sola serve per saltare una riga.
Attenzione che la parola PRINT seguita da # (detto cancelletto o diesis), PRINT #, indica una istruzione di uscita su cassetta o dischetto. È quindi tutta un'altra cosa.

INPUT

Anche INPUT è simile in tutti i dialetti BASIC. Dopo la parola INPUT si può mettere un messaggio che deve essere tra virgolette. Poi seguono le variabili a cui attribuire dei valori in ingresso.
Sin dalla preistoria del BASIC l'istruzione INPUT ha sempre visualizzato un punto interrogativo per avvisare che il calcolatore attende dei dati in ingresso. Questo punto interrogativo appare anche se si usa un messaggio, molto più comodo e professionale.

```
100 INPUT "DARE NOME E COGNOME"; A$,B$

DARE NOME E COGNOME? SERGIO, ROSSI
```

In alcuni calcolatori è possibile inibire l'apparizione del punto interrogativo se dopo il messaggio si usa una virgola invece di un punto e virgola.
Attenzione che il calcolatore è molto severo sulla correttezza dei dati forniti in risposta a INPUT. Se confondete variabili numeriche con quelle di stringa, o date meno dati di quelli richiesti, il calcolatore ripete la domanda, ma può anche interrompere il programma. Ricordate poi una regola molto importante. Se nella INPUT è indicata una variabile di stringa, potete dare in ingresso qualunque carattere, lettere o numeri:

```
100 INPUT DA$

? PROVA111PROVA222***
```

Se invece la variabile è numerica potete dare solo numeri (anche in formato esponenziale):

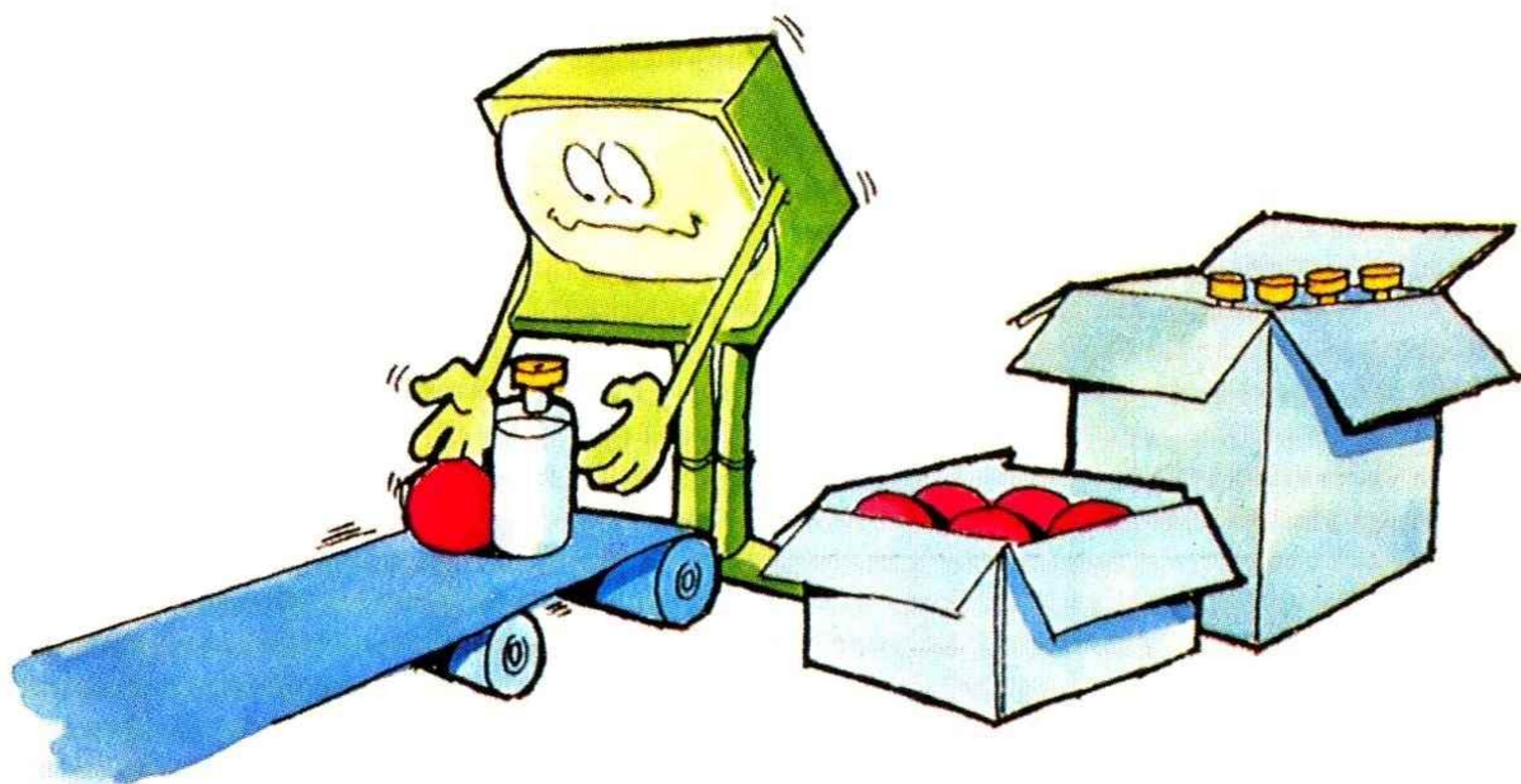
```
100 INPUT M, N

? 23456.55 , 3.14 E 2

3.14 E 2 è il formato esponenziale di 314.
```

In alcuni dialetti BASIC esistono anche altre istruzioni che permettono l'ingresso di dati in modo forse più complicato, ma più sicuro: GET, INKEY$, INPUT$. Avremo in seguito occasione di usare anche queste istruzioni.

DATA READ e RESTORE

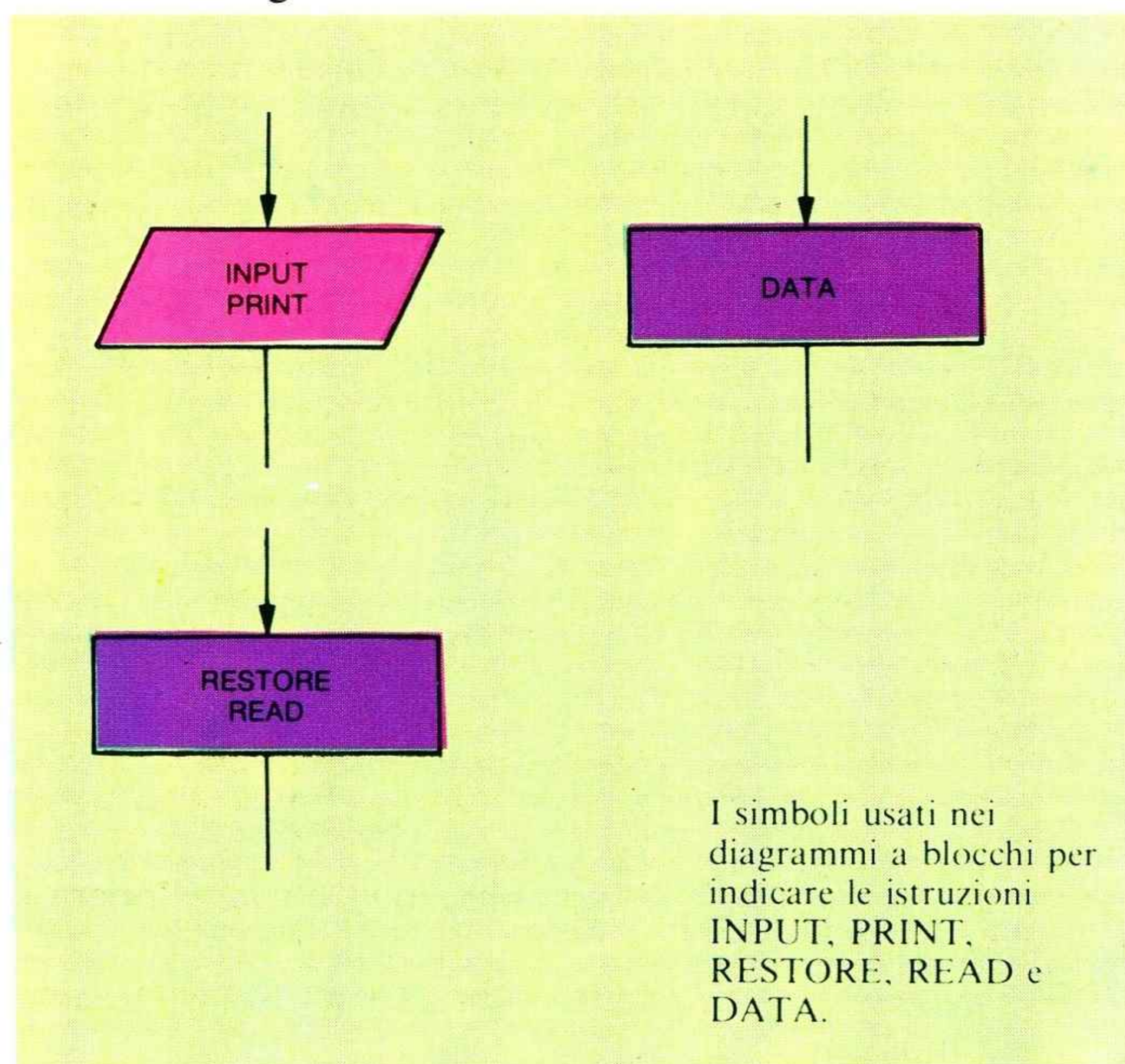
Queste istruzioni, (derivate dall'uso delle schede perforate nei calcolatori e vecchie quanto il BASIC) sono sicuramente quelle che presentano meno varianti.
Con DATA si inseriscono i dati (non molti) direttamente nel programma; con READ si associano alle variabili; RESTORE serve solo per poter ricominciare tutto da capo.
Un esempio molto classico dell'impiego di DATA e READ è quello di avere nel programma i mesi dell'anno e di richiamarli quando servono. Nel programma seguente si chiede il numero del mese e il calcolatore stampa il mese corrispondente in lettere:

```
100 DATA GENNAIO, FEBBRAIO, MARZO,
    APRILE
110 DATA MAGGIO, GIUGNO, LUGLIO,
    AGOSTO
120 DATA SETTEMBRE, OTTOBRE,
    NOVEMBRE, DICEMBRE
130 INPUT "QUALE MESE"; M
140 FOR K = 1 TO M
150 READ P$
160 NEXT K
170 PRINT P$
180 RESTORE
190 GOTO 130
RUN
QUALE MESE? 6
GIUGNO
```

Usare una sola DATA o più istruzioni è la stessa cosa (istruzioni 100, 110 e 120). Il ciclo FOR NEXT legge con READ i mesi sino a quello voluto (variabile numerica M). All'uscita dal ciclo P$ contiene il mese scelto che viene quindi stampato. RESTORE riporta il puntatore delle DATA all'inizio per ripetere eventualmente il programma.
Una applicazione di DATA più complessa permette di costruire dei file direttamente in un programma. In BASIC è molto facile cambiare una linea di programma e lo si può fare anche in modo interattivo.
La linea con DATA funziona in questo caso come un piccolo file molto utile in quei calcolatori che possono salvare su cassette solo i programmi e non i file di dati.

Il metodo è il seguente. Scrivere tutto il programma completo anche delle READ per leggere le DATA, ma non inserire le DATA salvo lasciare liberi i numeri di linea in cui dovrebbero essere scritte. Per esempio si possono lasciare liberi dei numeri molto alti come 1000, 1010, ecc.. Prima delle istruzioni che contengono READ inserire delle istruzioni come le seguenti:

```
460 ..
470 PRINT "FILE DATA GIA' CARICATO"
480 INPUT Y$
490 IF Y$ = "SI" GOTO 540
500 PRINT "BATTERE LE LINEE DI
    PROGRAMMA DELLE DATA"
510 PRINT "AD INIZIARE DALLA LINEA
    1000."
520 PRINT "POI DARE IL COMANDO SAVE"
530 STOP
540 ... Qui continua il programma
550 ... con le istruzioni READ
560 ...
1000 DATA ...
1010 DATA ...
```

Succede così che il programma dapprima chiede se avete già inserito le DATA. Se non lo avete già fatto le battete dopo lo STOP e poi salvate tutto il programma compresi i nuovi dati inseriti nelle DATA. A questo punto potete dare il RUN definitivo e lasciare che le READ usufruiscano dell'archivio DATA. Ovviamente quando richiamerete il programma con LOAD, caricherete tutto anche i dati che potrete lasciare come sono oppure cambiare con una tecnica analoga.

I simboli usati nei diagrammi a blocchi per indicare le istruzioni INPUT, PRINT, RESTORE, READ e DATA.

Coniugazione dei verbi

Vedere un calcolatore che elabora dei numeri ed esegue dei calcoli matematici forse non meraviglia nessuno. È ovvio che un calcolatore faccia dei conti! Se invece vediamo un computer "elaborare" un testo o coniugare dei verbi, ci sembra che abbia qualcosa di magico.
Il programma che presentiamo coniuga il presente indicativo dei verbi che finiscono in "are", come camminare o guardare. Vedremo che ciò è possibile grazie ad un semplice programma in BASIC.
Elaborare delle parole vuol dire modificare delle stringhe di caratteri. Le parole, infatti, non sono altro che successioni di caratteri, a cui noi attribuiamo un significato, che soddisfano le regole della grammatica.
Sappiamo già che il BASIC, come la maggior parte dei linguaggi di programmazione, può elaborare le stringhe con istruzioni come:

```
MID$
LEFT$
RIGHT$
INSTR
LEN
```

o l'operatore unione "+".
A questo punto se conosciamo le regole grammaticali (l'algoritmo), possiamo tradurle in un programma, impiegando appunto queste istruzioni del BASIC.
La possibilità di scrivere questi programmi ha però delle limitazioni dovute alle irregolarità di tutte le lingue parlate. La regola che abbiamo tradotto in BASIC, per coniugare il presente indicativo dei verbi della prima coniugazione (quelli che finiscono in "-are", per intenderci) non va bene se il verbo è irregolare, come "fare", oppure richiede desinenze diverse, come "mangiare". Infatti un verbo regolare e semplice viene così coniugato:

Io cammino
Tu cammini
Egli cammina
Noi camminiamo
Voi camminate
Essi camminano

Si vede che le desinenze che dobbiamo usare sono:

— o
— i
— a
— iamo
— ate
— ano

Queste desinenze, che sono inserite nella DATA all'istruzione 540, devono essere attaccate alla radice "cammin-" ottenuta da "cammin-are".
Queste regole vanno bene per la maggior parte dei verbi della prima coniugazione e potrebbero essere estese alle altre coniugazioni.
Un breve inciso. Esercizi di BASIC come questo non sono delle raffinatezze prive di utilità. Infatti, una delle più interessanti applicazioni dei computer, assieme ai programmi di word processing (elaborazione di testi), è quella in cui un testo battuto liberamente viene "passato alla correzione" da parte del computer. Un computer può:

- controllare la suddivisione in sillabe quando si va a capo;
- controllare la coniugazione dei verbi, in particolare dei congiuntivi e condizionali che sono spesso sbagliati;
- controllare l'esatta battitura delle parole, perlomeno delle più usuali;
- controllare la punteggiatura.

Tutte queste cose non vogliono dire che il computer "capisce le parole" (semantica), ma solo che una frase d'italiano battuta alla tastiera può essere corretta automaticamente a livello ortografico o grammaticale, da un computer.
Anche se pochi dei controlli menzionati sono oggi possibili con un personal, questo può aiutarci se non altro a scrivere più correttamente.

Il programma

La struttura del programma è abbastanza semplice e lineare. Nella prima parte vengono visualizzati dei messaggi sullo schermo poi, dopo una conferma d'inizio, vengono caricati (con le istruzioni READ e DATA) i due vettori PR$ e D$ che contengono i pronomi e le desinenze.
A questo punto il programma chiede (con INPUT) di quale verbo si vuole la coniugazione e controlla se è un verbo che finisce in "are" (con l'istruzione IF THEN e la funzione RIGHT$).
La visualizzazione del presente indicativo avviene molto semplicemente con un ciclo FOR NEXT e con PRINT.
Il programma prima di terminare chiede se si vuole continuare con un altro verbo.
Ecco la spiegazione dettagliata delle singole istruzioni:
100-140: Istruzioni di commento REM per dare un titolo al listato.
150, 160, 220, 280, ecc.: Sono linee di programma in cui compare solo un carattere (:), che permette di lasciare le linee vuote per separare in blocchi il listato, al fine di facilitarne la lettura. Se avessimo dato solo un numero di linea, il BASIC non lo avrebbe accettato. Se un personal non riconosce i due punti (:) come separatore d'istruzioni, si può sempre mettere REM e basta.
170: Pulitura iniziale dello schermo: con l'istruzione HOME per l'Apple, PRINT con SHIFT CLR/HOME per il Commodore 64 e CLS per lo Spectrum. Questa istruzione iniziale è molto importante per far partire un programma senza residui precedenti sullo schermo.
180-260: Sono messaggi che visualizzano sullo schermo il contenuto del programma e chiedono una conferma d'inizio del programma stesso. Tutti i programmi dotati di una certa professionalità dovrebbero iniziare solo se viene dato un comando di conferma.
270: Conferma dell'inizio. La variabile C$ è fittizia (dummy). L'istruzione INPUT ferma l'esecuzione, se non vien battuto un valore. Diversamente si può scrivere:

```
270 C$ = INKEY$: IF C$ < > " " GOTO 270
```

290:l'istruzione RESTORE in questo punto non è strettamente necessaria, ma è raccomandabile porre sempre RESTORE prima di usare i valori contenuti nelle DATA, per essere sicuri che il BASIC prelevi dall'inizio i valori DATA. RESTORE è particolarmente importante, se si deve rieseguire un programma (senza usare RUN) con GOTO. Ricordiamo che RESTORE riporta all'inizio il puntatore dell'archivio delle DATA.
300-350: Il vettore PR$ ( ) viene caricato con i pronomi; il vettore D$ ( ) con le desinenze.
370: Lo schermo viene nuovamente pulito.
380-420: In ingresso viene chiesto un verbo della prima coniugazione ("are") e vien fatto un controllo sulla desinenza (IF RIGHT$ alla linea 420).
440: V$ contiene la radice del verbo ottenuta con LEFT$ e staccando "are".
450-480: Un semplice ciclo FOR NEXT con delle istruzioni PRINT visualizzano il presente indicativo. SPC(5) alla linea 470 serve solo per spostare i verbi a destra di 5 caratteri. La congiunzione tra la radice e le desinenze viene fatta sfruttando la proprietà di PRINT di non andare a capo, se si usa il punto e virgola (;). Altrimenti per concatenare la stringa V$ con D$ si sarebbe dovuto usare l'operatore + (& in altri BASIC).
490-510: Viene chiesto se continuare, oppure far terminare il programma.

530-540: Sono le DATA che fungono da archivio dei pronomi e delle desinenze.

Commenti

Se proviamo a coniugare "mangiare" con questo programma, ci accorgiamo di commettere qualche errore:

```
Io mangio
Tu mangii
Egli mangia
Noi mangiiamo
Voi mangiate
Essi mangiano
```

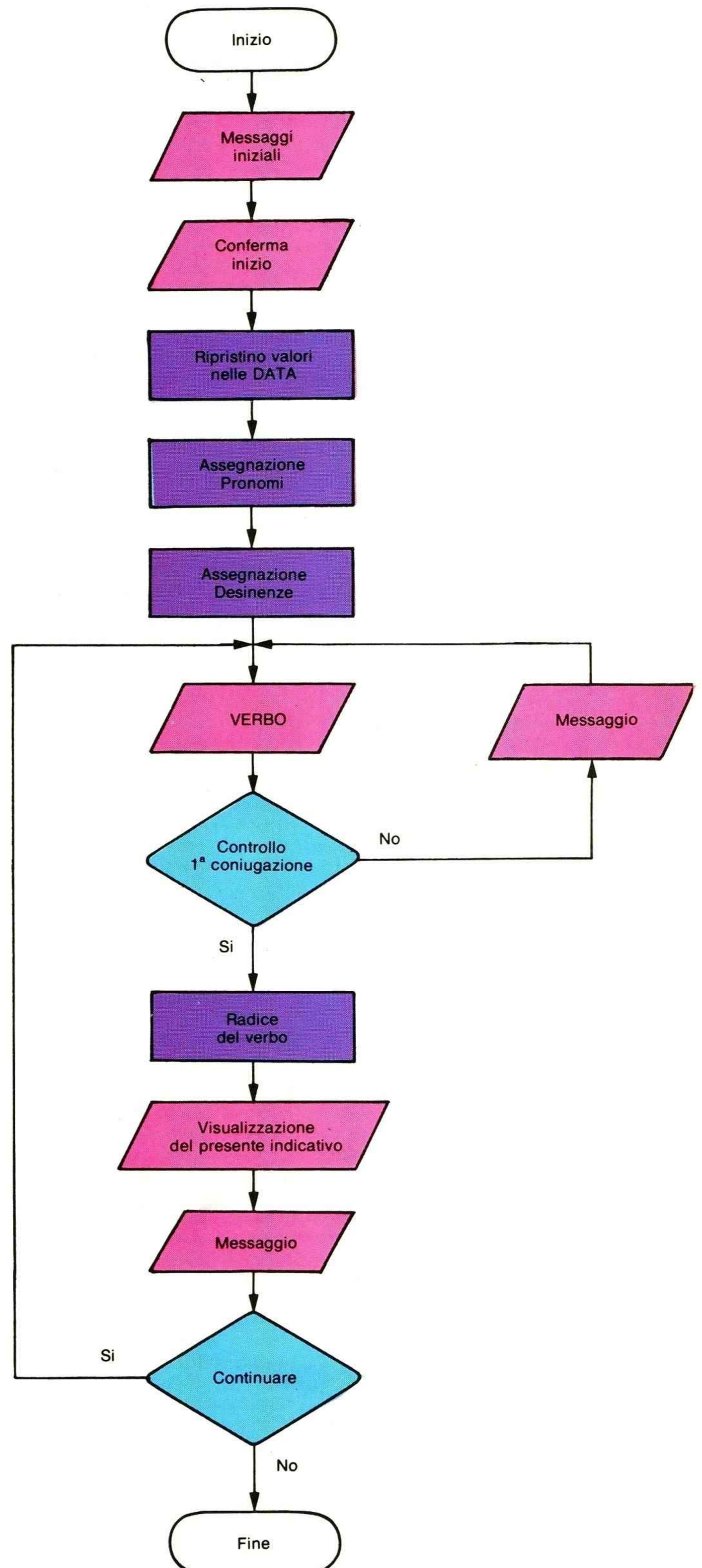
Come correggere questi errori? Il programma dovrebbe fare degli ulteriori controlli. Nel caso di "mangiare" potrebbe vedere se è presente una "i" che precede "are" e cambiare le desinenze in: io, i, ia, iamo, iate, iano. Più semplicemente, prima di visualizzare una stringa si potrebbe cercare se esiste una coppia di "ii" e sostituirla con una sola "i":

mangiiamo con mangiamo

```
460 FOR K = 1 TO 6
470 PRINT SPC(5);
472 VC$ = V$ + D$(K): IN = INSTR (VC$, "II")
474 IF IN <> 0
    THEN VC$ = LEFT$ (VC$, IN - 1) +
    MID$ (VC$, IN + 1)
476 PRINT PR$(K); " "; VC$
480 NEXT K
```

Provate a fare questa modifica. Se non avete la funzione INSTR, che cerca una stringa all'interno di un'altra (INSTR dà il valore zero se la stringa interna manca), vi lasciamo scrivere una routine per sostituirla (più avanti vi daremo la risposta).

Tabella delle variabili

C$	Variabile fittizia per confermare dell'inizio del programma.
PR$ ( )	Vettore dei pronomi.
D$ ( )	Vettore delle desinenze.
VB$	Verbo di cui si vuole la coniugazione.
V$	Radice del verbo VB$.
F$	Variabile per conferma ripetizione del programma.

```
100  REM  ******************************
     ******
110  REM  *
          *
120  REM  *  PROGRAMMA CONIUGAZIONE VER
     BI   *
130  REM  *
          *
140  REM  ******************************
     ******
150 :
160 :
170  HOME : REM  PULISCE LO SCHERMO
180  PRINT "    QUESTO PROGRAMMA PERMETT
     E"
190  PRINT "    DI CONIUGARE IL PRESENTE
     "
200  PRINT "    INDICATIVO DEI VERBI REG
     OLARI"
210  PRINT "    DELLA PRIMA CONIUGAZIONE
     "
220 :
230  PRINT
240  PRINT "    IN INGRESSO BISOGNA DARE
     "
250  PRINT "    IL VERBO ALL'INFINITO"
260  PRINT : PRINT : PRINT "    PER INIZ
     IARE PREMERE RETURN": PRINT
270  INPUT C$:
280 :
290  RESTORE
300  FOR J = 1 TO 6
310  READ PR$(J)
320  NEXT J
330  FOR K = 1 TO 6
340  READ D$(K)
350  NEXT K
360 :
370  HOME : REM  PULISCE LO SCHERMO
380  PRINT : PRINT : PRINT "DARE UN VER
     BO ALL'INFINITO : ": PRINT
390  INPUT VB$
400 :
410  REM  CONTROLLO PRIMA CONIUGAZIONE
420  IF  RIGHT$ (VB$,3) <  > "ARE" THEN
      PRINT "NON VA BENE": GOTO 370
430 :
440 V$ =  LEFT$ (VB$, LEN (VB$) - 3)
450  PRINT : PRINT : PRINT "ECCO IL PRE
     SENTE INDICATIVO :": PRINT : PRINT

460  FOR K = 1 TO 6
470  PRINT  SPC( 5);PR$(K);"  ";V$;D$(K
     )
480  NEXT K
490  PRINT : PRINT : INPUT "VUOI CONTIN
     UARE ? <S> ";F$
500  IF F$ = "S" THEN 370
510  END
520 :
530  DATA  IO, TU, EGLI, NOI, VOI, ESSI

540  DATA  O, I, A, IAMO, ATE, ANO
```

```
    QUESTO PROGRAMMA PERMETTE
    DI CONIUGARE IL PRESENTE
    INDICATIVO DEI VERBI REGOLARI
    DELLA PRIMA CONIUGAZIONE

    IN INGRESSO BISOGNA DARE
    IL VERBO ALL'INFINITO


    PER INIZIARE PREMERE RETURN

?

DARE UN VERBO ALL'INFINITO :

?CAMMINARE


ECCO IL PRESENTE INDICATIVO :

     IO  CAMMINO
     TU  CAMMINI
     EGLI  CAMMINA
     NOI  CAMMINIAMO
     VOI  CAMMINATE
     ESSI  CAMMINANO


VUOI CONTINUARE ? <S> S

DARE UN VERBO ALL'INFINITO :

?SALTARE


ECCO IL PRESENTE INDICATIVO :


     IO  SALTO
     TU  SALTI
     EGLI  SALTA
     NOI  SALTIAMO
     VOI  SALTATE
     ESSI  SALTANO


VUOI CONTINUARE ? <S> S


DARE UN VERBO ALL'INFINITO :

?GUARDARE


ECCO IL PRESENTE INDICATIVO :


     IO  GUARDO
     TU  GUARDI
     EGLI  GUARDA
     NOI  GUARDIAMO
     VOI  GUARDATE
     ESSI  GUARDANO


VUOI CONTINUARE ? <S> N
```

Indovina le parole

Ecco un utile esercizio. Nulla di difficile, ma un esempio di come si possa scrivere un programma che abbia tutti gli elementi per costituire un gioco: scegliere un livello di difficoltà, porre delle domande, attendere delle risposte, cambiare le videate del gioco, contare gli errori, dare la risposta esatta se non viene trovata oppure avvisare della vittoria. Non abbiamo usato della grafica per non complicare il programma. Le immagini in colore o grafiche possono essere inserite in qualunque punto del programma per arricchire le videate e creare effetti particolari.
Il programma invita il giocatore a indovinare delle parole per tentativi: alla tastiera viene battuta una lettera L$ e se questa è presente nella parola misteriosa sul video appare questa lettera inserita nel punto giusto (anche in più punti). Per aiutare il giocatore appare sul video una stringa (T$) con tanti puntini quante sono le lettere della parola misteriosa W$ (linea 360).
Le parole da indovinare sono già inserite nel programma alle linee da 770 a 790 con le istruzioni DATA. Una variante al programma potrebbe prevedere che i giocatori siano due. Il primo batte alla tastiera una parola, che l'altro non vede e che poi tenta di indovinare. Il gioco si alterna e vince chi indovina più parole. Per indovinare una parola si può scegliere una lettera un certo numero di volte: P calcolato come 11 − D; D è il livello di difficoltà (linea 290) e 11 è un numero arbitrario scelto da noi.
Se il calcolatore è dotato di una funzione orologio, si potrebbe consentire un certo tempo per indovinare, scaduto il quale il giocatore perde.
Per scegliere una parola a caso tra quelle contenute nelle DATA si ricorre a un semplice trucco. Sapendo quante sono le parole disponibili (variabile M che riceve il valore dalla DATA della linea 760), si genera dapprima un numero a caso (N) tra 1 e M, poi si esegue un ciclo FOR NEXT di lettura delle parole, con READ, sino a N.

Il programma

Il programma prende a caso una parola dell'archivio DATA e visualizza una stringa di puntini di pari lunghezza. Dopo aver stabilito il livello di difficoltà chiede una lettera. Se questa lettera è presente nella parola la visualizza e la toglie dalla stringa alfabeto. Per ogni errore viene tolto un punto. Il programma prosegue sino a che vi sono punti disponibili.
100-120: Titolo e commenti iniziali.
170: Pulitura iniziale dello schermo: con l'istruzione HOME per l'Apple, PRINT con SHIFT CLR/HOME per il Commodore 64 e CLS per lo Spectrum. Questa istruzione iniziale è molto importante per far partire un programma senza residui precedenti sullo schermo.
180-240: Visualizzazione dei messaggi iniziali e conferma dell'inizio del gioco.
260: RESTORE è necessaria quando il programma viene successivamente ripetuto.
280-320: Viene scelto un livello di difficoltà e da questo calcolato il numero di volte che si può tentare.
340-350: Scelta a caso di una parola tra quelle contenute nelle DATA.
360: T$ contiene la parola in risposta e viene inizializzata con tanti puntini quante sono le lettere della parola misteriosa.
370: Stringa AF$ contenente le lettere dell'alfabeto, a cui vengono tolte le lettere scelte dal giocatore.
400-450: Messaggi e visualizzazione del campo T$ e delle lettere disponibili AF$.
470-480: Ingresso della lettera scelta L$. Abbiamo posto il controllo solo se non viene data nessuna lettera e battuto RETURN. Si sarebbe potuto porre un controllo più forte per eliminare battute di più lettere o di caratteri non letterali:

```
482 IF LEN(L$) > 1 THEN 470
484 IF (ASC(L$) < 65) OR (ASC(L$) > 90)
    THEN 470
```

La seconda istruzione IF THEN controlla se viene data una lettera maiuscola; nello stesso modo si potrebbe controllare se la lettera è minuscola.
490-520: Controllo della presenza della lettera nella stringa AF$ delle lettere disponibili.
550-570: Ricerca della lettera nella parola misteriosa (misteriosa ... per modo di dire!). Se presente, il flag F è posto a 1.
590: Controllo se la parola è stata indovinata.
600: Se la lettera era sbagliata, viene diminuito il numero P delle prove.
610-660: Se non sono più possibili delle prove la partita è persa.
680-710: Messaggio di vincita e visualizzazione della parola giusta.
720-730: Conferma continuazione del gioco.
760-790: Archivio delle DATA.

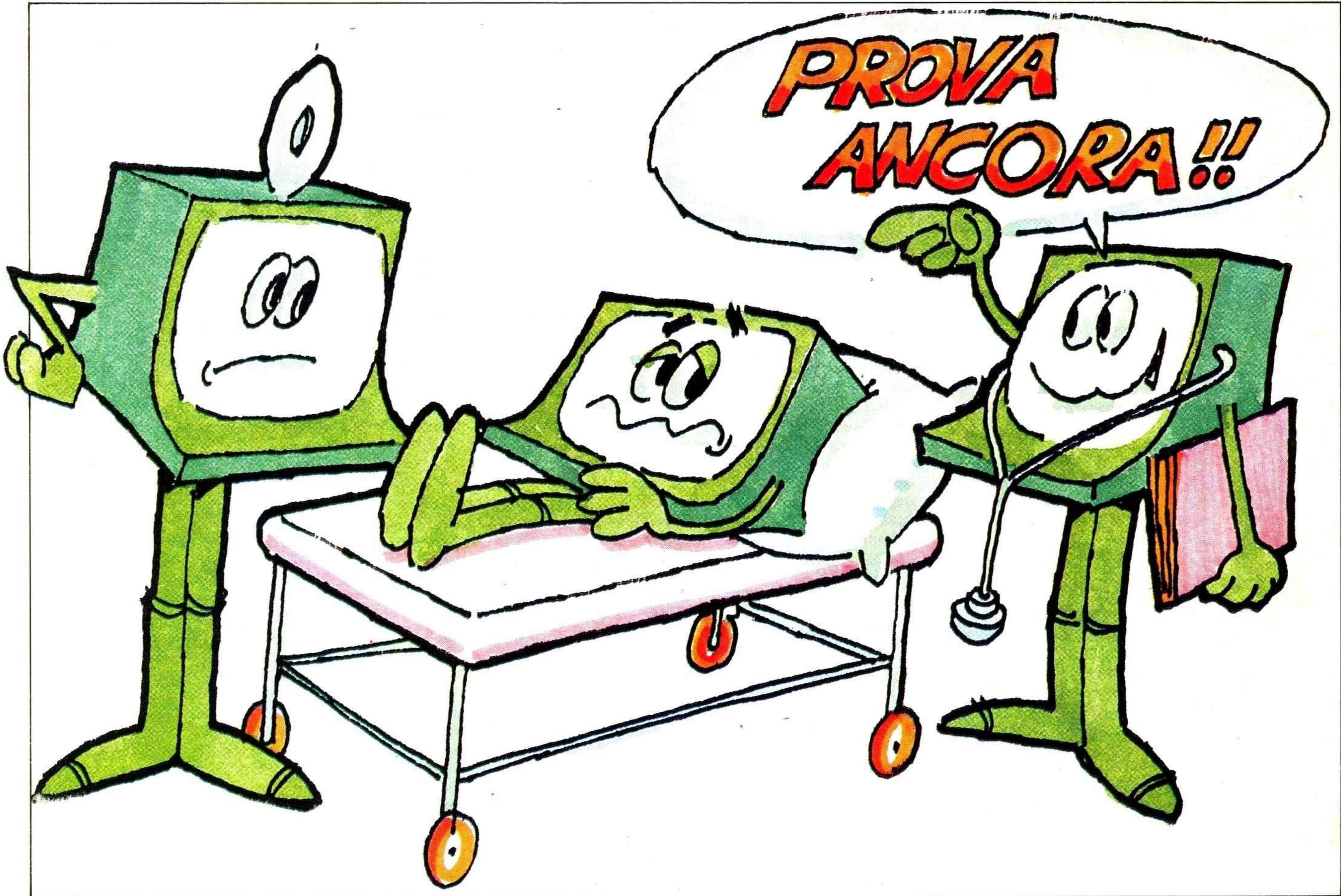
Inizio
Messaggi iniziali
B
Pulisce schermo
Scelta livello difficoltà fra 1 e 5
No
Livello valido?
Si
Numero prove consentite
Scelta di una parole a caso dalle DATA
Campo che conterrà la parola da indovinare
Alfabeto
D
Messaggio: prove consentite
Visualizzazione campo parola misteriosa
A
A
Lettere disponibili
Scelta di una lettera
Ripetere se non viene data una lettera
Si
No
Ricerca della lettera nella parola misteriosa e inserimento nella stringa campo
Parola indovinata
No
C
Si
Messaggio: vincita
Parola giusta
Messaggio: continuare
Si?
Si
B
No
Fine
C
Lettera singola indovinata
No
Diminuisce prove
Si
Ancora prove?
Si
D
No
Messaggio: perdita

```
100  REM ***********************
110  REM   INDOVINA LE PAROLE
120  REM **********************
150  :
160  :
170  HOME
180  PRINT : PRINT : PRINT
190  PRINT "        QUESTO PROGRAMMA VI INVITA
     "
200  PRINT "        A INDOVINARE DELLE PAROLE"
210  PRINT "        DOPO AVER SCELTO UN LIVELL
     O"
220  PRINT "        DI DIFFICOLTA'"
230  PRINT : PRINT
240  INPUT "  PREMERE RETURN PER INIZIARE ";
     Y$
250  :
260  HOME : RESTORE
270  :
280  PRINT : PRINT
290  PRINT " SCEGLIERE UN LIVELLO DI DIFFICO
     LTA' TRA 1 E 5"
300  PRINT : INPUT D
310  IF (D < 1) OR (D > 5) THEN 260
320 P = 11 - D: REM  PROVE CONSENTITE
330  :
340  READ M:N =  RND (1) * M + 1
350  FOR I = 1 TO N: READ W$: NEXT I
360 T$ =  LEFT$ ("...............",  LEN (W$
     ))
370 AF$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
380  :
390  REM  ELABORAZIONE
400  HOME : VTAB 5
410  PRINT "PUOI PROVARE  ";P;" VOLTE"
420  PRINT : PRINT T$
430  PRINT : PRINT "LETTERE CHE PUOI PROVARE
     : "
440  PRINT
450  PRINT AF$
460  :
470  PRINT : INPUT "QUALE LETTERA ? ";L$
480  IF L$ = "" THEN 470
490  FOR I = 1 TO  LEN (AF$)
500  IF  MID$ (AF$,I,1) = L$ THEN AF$ =  MID$
     (AF$,1,I - 1) +  MID$ (AF$,I + 1): GOTO
     540
510  NEXT I
520  PRINT : PRINT "  LETTERA GIA' SCELTA. R
     IPETERE: ": GOTO 470
530  :
540  REM LETTERE DELL'ALFABETO
550 F = 0: FOR I = 1 TO  LEN (W$)
560  IF  MID$ (W$,I,1) = L$ THEN F = 1:T$ =
      MID$ (T$,1,I - 1) + L$ +  MID$ (T$,I +
     1)
570  NEXT I
580  :
590  IF T$ = W$ THEN 680
600  IF F = 0 THEN P = P - 1
610  IF P > 0 THEN 390
620  :
630  REM  PERDITA
640  PRINT : PRINT
650  PRINT : PRINT "  HAI PERSO...LA PAROLA
     ERA:"
660  GOTO 710
670  :
680  REM  VINCITA
690  PRINT : PRINT : PRINT " BRAVO HAI TROVA
     TO LA PAROLA GIUSTA! "
700  PRINT
710  PRINT : PRINT "              --";W$;"--"
720  PRINT : INPUT "   VUOI CONTINUARE ? (S)
     ";R$
730  IF R$ = "S" THEN 260
740  END
750  :
760  DATA  10
770  DATA  QUASAR,ATOLLO,CAVALLO
780  DATA  PERSONAL,ABC,BASIC,ENTROPIA
790  DATA  PILOTA,ARCHIVIO,BYTE
```

```
       QUESTO PROGRAMMA VI INVITA
       A INDOVINARE DELLE PAROLE
       DOPO AVER SCELTO UN LIVELLO
       DI DIFFICOLTA'


  PREMERE RETURN PER INIZIARE


 SCEGLIERE UN LIVELLO DI DIFFICOLTA' TRA 1 E 5

?4
PUOI PROVARE  7 VOLTE

......

LETTERE CHE PUOI PROVARE :

ABCDEFGHIJKLMNOPQRSTUVWXYZ

QUALE LETTERA ? A
PUOI PROVARE  7 VOLTE

A.....

LETTERE CHE PUOI PROVARE :

BCDEFGHIJKLMNOPQRSTUVWXYZ

QUALE LETTERA ? M
PUOI PROVARE  6 VOLTE

A.....

LETTERE CHE PUOI PROVARE :

BCDEFGHIJKLNOPQRSTUVWXYZ

QUALE LETTERA ? L
PUOI PROVARE  6 VOLTE

A..LL.
```

```
LETTERE CHE PUOI PROVARE :

BCDEFGHIJKNOPQRSTUVWXYZ

QUALE LETTERA ? O
PUOI PROVARE  6 VOLTE

A.OLLO

LETTERE CHE PUOI PROVARE :

BCDEFGHIJKNPQRSTUVWXYZ

QUALE LETTERA ? T

 BRAVO HAI TROVATO LA PAROLA GIUSTA!

        --ATOLLO--

   VUOI CONTINUARE ? <S> N
```

Tabella delle variabili

Y$	Variabile fittizia per conferma dell'inizio del programma.
D	Livello di difficoltà.
M	Numero di parole contenute nell'archivio DATA.
N	Numero a caso per scegliere la parola da indovinare.
W$	Parola da indovinare.
T$	Stringa di puntini di pari lunghezza della parola da indovinare.
AF$	Stringa contenente le lettere dell'alfabeto. Da questa stringa vengono tolte le lettere scelte dopo ogni tentativo.
P	Totale delle prove possibili.
L$	Lettera scelta.
F	Flag per indicare se la lettera scelta era presente nella parola misteriosa.
R$	Variabile per confermare la ripetizione del gioco.

I Word Processor

Programmare ... non programmare.
Fare da se o impiegare un programma già pronto - che dilemma!
Gli entusiasti del "chi fa da se fa per tre" preferiscono scrivere tutti i programmi di cui hanno bisogno. Sia il programmino tipo tabellina pitagorica, sia che si tratti di una grande contabilità o gestione magazzini.
Il giusto sta sempre nel mezzo. Scrivere un programma per database può richiedere molti anni-uomo. È inutile iniziare per poi arenarsi in un megapasticcio tipo "programma spaghetti" infarcito di GOTO.
I programmi professionali, i package, ci sono già belli e pronti, funzionali e anche economici.
All'opposto non si deve pretendere di trovare in negozio programmini elementari tipo problema di fisica della vasca da bagno che si riempie.
L'ideale è sfruttare il calcolatore per le cose migliori che ci offre.
Programmare da soli è un piacevole hobby, che ci rende anche più competenti e in grado di apprezzare la macchina e il lavoro fatto da altri. Parimenti la conoscenza dei package di software più comuni e importanti è altrettanto indispensabile.
I package professionali hanno tre principali applicazioni a cui se ne è aggiunta di recente una quarta: word processor (elaborazione di testi), database (archivi di dati), spreadsheet (fogli elettronici) e communications (programmi per comunicazioni tra computer).
Ogni gruppo di pacchetti ha i suoi leader; programmi molto noti e famosi o addirittura capostipiti come VisiCalc tra i fogli elettronici.
Dei primi tre gruppi faremo una presentazione e daremo la descrizione dettagliata dei package più importanti.

In qualunque tipo di attività è indispensabile scrivere: dover redarre lettere, appunti, relazioni è all'ordine del giorno per tutti.
Naturale, quindi, che si sia cercato di rendere più agevole questo tipo di attività sfruttando la potenza e la flessibilità di impiego dei personal computer. Sono così nati i word processor, che sono dei programmi che consentono di utilizzare i personal computer come sofisticate macchine per scrivere. I word processor, ossia i sistemi di videoscrittura o di elaborazione testi, differiscono un po' a seconda del calcolatore cui sono destinati, ma svolgono in definitiva la medesima funzione: permettono all'uti-

L'ufficio è uno dei luoghi di lavoro in cui si apprezzano maggiormente i vantaggi legati all'utilizzo di un sistema di elaborazione testi.

lizzatore di scrivere direttamente sulla tastiera il testo desiderato, facendolo apparire sul video, per poi stamparlo su carta secondo un formato scelto liberamente.

Nel seguito vedremo più approfonditamente come lavorano questi programmi: ora notiamo come una caratteristica comune a tutti sia l'estrema semplicità d'uso. In pratica chiunque è capace di usare una normale macchina per scrivere, è senz'altro in grado di utilizzare un word processor. I risultati saranno sicuramente positivi, e ben pochi vorranno tornare al vecchio sistema!

Il fatto che sia molto facile utilizzare i sistemi di videoscrittura, non significa che i relativi programmi siano altrettanto elementari: si tratta in realtà di programmi molto complessi di cui l'utente usufruisce in modo semplice e immediato. Spesso in informatica si verifica che per facilitare l'utilizzo di un programma, occorre complicare notevolmente il compito del programmatore.

L'introduzione del testo

Quando si lancia l'esecuzione di un programma di elaborazione testi, viene normalmente presentato lo schermo completamente privo di caratteri e il cursore (generalmente un rettangolino o un trattino), che segnala la posizione in cui verrà visualizzato il successivo carattere digitato.

Questo è il nostro "foglio": è un foglio elettronico, (il video) ma che possiamo utilizzare proprio come se fosse un normale foglio di carta inserito nella macchina per scrivere. Proprio come se dovessimo scrivere a macchina quindi, possiamo battere il nostro testo, spostare il cursore in qualunque posizione dello schermo (simulando quindi il movimento del carrello), e in più possiamo fare molte altre cose. Innanzitutto possiamo manipolare a piacere anche in fasi successive il testo già scritto: cancellare, inserire o modificare singoli caratteri o addirittura intere parole e frasi è reso semplice e immediato dal fatto che il testo che viene presentato su video è in realtà memorizzato provvisoriamente nella RAM, la memoria a lettura e scrittura del calcolatore.

Il testo come appare su carta in funzione dei codici inseriti. Da notare che i comandi che appariranno all'inizio del testo su video, qui non sono riportati.

```
In qualunque tipo di attivita' e' indispensabile scrivere:
dover redarre lettere, appunti, relazioni e' all'ordine del
giorno per tutti.
Naturale, quindi, che si sia cercato di rendere piu' agevole
questo tipo di attivita' sfruttando la potenza e la
flessibilita' di impiego dei personal computer. Sono cosi'
nati i Word Processor, che sono dei programmi che consentono
di utilizzare i personal computer come sofisticate macchine
per scrivere. I Word Processor, ossia i sistemi di
videoscrittura o di elaborazione testi, differiscono un po'
a seconda del calcolatore cui sono destinati, ma svolgono in
definitiva la medesima funzione: permettono all'utilizzatore
di scrivere direttamente dalla tastiera, facendolo apparire
sul video, il testo desiderato, per poi stamparlo su carta
secondo un formato scelto liberamente.
Nel seguito vedremo piu' approfonditamente come lavorano
questi programmi: ora notiamo come una caratteristica comune
a tutti sia l'estrema semplicita' d'uso. In pratica chiunque
e' capace di usare una normale macchina per scrivere, e'
senz'altro in grado di utilizzare un Word Processor. I
risultati saranno sicuramente positivi, e ben pochi vorranno
tornare al vecchio sistema!
Il fatto che sia molto facile utilizzare i sistemi di
videoscrittura, non significa che i relativi programmi siano
altrettanto elementari: si tratta in realta' di programmi
molto complessi di cui l'utente usufruisce in modo semplice
e immediato. Spesso in informatica si verifica che per
facilitare l'utilizzo di un programma, occorre complicare
notevolmente il compito del programmatore.
```

Ecco come si presenta il video con il testo da stampare. Si tratta del programma Applewriter IIe funzionante sul personal computer Apple IIe. Da notare all'inizio del testo i tre comandi lm rm fj preceduti da un puntino: si tratta dei codici che determinano l'impaginazione su carta del testo fissando rispettivamente il margine sinistro, destro e causando la giustificazione destra e sinistra.

Le modifiche necessarie in memoria vengono eseguite in modo completamente automatico dal programma, che provvede anche ad aggiornare il testo sul video.
Queste possibilità di intervento danno già una idea precisa della comodità di impiego di un sistema di videoscrittura: non è più necessario scrivere una minuta del testo per poi batterlo in bella copia, in quanto lo stesso scritto iniziale, dopo le modifiche desiderate, diventa quello definitivo.
Quando si batte un testo a video, non bisogna generalmente preoccuparsi della marginatura e della impaginazione: lo scopo, a questo punto, è quello di stabilire cosa si vuol scrivere e non come debba apparire su carta quanto scritto (vedremo che questa seconda esigenza sarà curata in un semplicissimo passo successivo) in questo modo l'utente può concentrarsi maggiormente sui contenuti del proprio testo.

La memorizzazione su disco

Prima di passare ad esaminare come avviene la stampa su carta del testo memorizzato, notiamo un'altra importante caratteristica dei word processor. Tutti i migliori sistemi di videoscrittura, permettono di memorizzare in modo permanente su un supporto magnetico (normalmente un disco), qualunque testo sia stato inserito. Ovviamente è possibile il passaggio inverso, cioè leggere da disco e far apparire sul video uno scritto precedentemente memorizzato.
Questa caratteristica è comodissima in molti casi: innanzitutto permette di scrivere e modificare uno stesso testo anche in tempi successivi, secondo le esigenze dell'utente.
Ma soprattutto permette di crearsi una biblioteca di testi che si usano frequentemente ed elimina la noia e la perdita di tempo di dover continuamente riscrivere le stesse cose.
Chi, ad esempio, deve scrivere spesso delle lettere con un testo più o meno simile e in cui varino per esempio solo la data, il destinatario e altri pochi dati, potrà leggere da disco una copia vecchia e poi modificare soltanto i dati desiderati. Sempre nel campo della corrispondenza, molti word

Pacchetto	Produttore	Hardware necessario	Tipo di supporto	Funzioni matematiche
Wordstar 3.3	Micropro	CP/M MS-DOS e altri	Disco	
Peachtext	Peachtree	CP/M	Disco	
Apple Writer	Apple Computer	Apple	Disco	Si
Perfect Writer	Transam Microsystems	CP/M	Disco	
WordPlus-PC	Wego Computers	IBM PC	Disco	
Deskmaster 2	Kuma Computers	Epson HX20	Micro cassette	
Super Writer	ACT	Sirius	Disco	
Wordplex	G. W. Computers	CP/M	Disco	
Magic Wand	G. W. Computers	CP/M	Disco	
MS-Word with MS-Mouse	Microsoft	IBM PC	Disco	
Easy Script	Precision Software	Commodore 64	Disco	
XED	Precision Software	UNIX	Disco/Nastro	

In questa tabella sono riportati alcuni dei più diffusi word processor esistenti. Per ognuno viene specificato, tra l'altro, se è dotato di qualche possibilità matematica.
Non tutti questi programmi sono reperibili in Italia.

processor possono essere abbinati ad archivi di indirizzi: in questo modo è possibile far generare automaticamente al calcolatore una serie di circolari personalizzate sfruttando un'unica lettera standard, e automatizzando il procedimento visto prima.

I comandi di impaginazione

Da quanto detto, si capisce che, per avere una buona resa dal proprio word processor sia necessaria, oltre ovviamente alla stampante, anche una efficiente unità di memorizzazione dati.
Dopo aver inserito il testo desiderato, arriva finalmente il momento di stamparlo. A questo punto occorre specificare al calcolatore come si vuol impaginare il testo in funzione delle dimensioni del foglio e dei gusti personali.
In definitiva, utilizzando questi sistemi, è il testo che si adatta al foglio di carta e non viceversa. In questo modo è possibile stampare uno stesso testo su formati di carta differenti, senza doverlo ribattere, ma semplicemente specificando di volta in volta le dimensioni prescelte. Queste indicazioni vengono fornite al calcolatore, inframezzando al testo degli speciali codici che specificano come deve avvenire la stampa su carta. Questi codici, che differiscono da un sistema all'altro, sono visibili solo sul testo a video e non sulla copia cartacea. I più usati sono quelli che fissano le colonne delle marginature sinistra e destra, l'ampiezza dell'interlinea, le dimensioni, espresse in numero di linee, della pagina di stampa e il tipo di giustificazione. Questa ultima possibilità è tipica dei sistemi di videoscrittura: un testo può essere marginato semplicemente a sinistra, come quando viene battuto normalmente a macchina. Inoltre è possibile la centratura (molto utile ad esempio per i titoli) e la marginatura completa, sinistra e destra, che genera un effetto molto simile a quello di una pagina stampata. La marginatura completa viene effettuata automaticamente dal sistema che provvede ad inserire degli spazi bianchi lungo tutta la riga, fra una parola all'altra, in modo da rendere fissa la lunghezza delle linee.

La stampa

Molti sistemi consentono di vedere su video come apparirà poi su carta un stampa in funzione dei codici inseriti: in questo modo è possibile fare velocemente delle prove e modificare i vari codici fino a ottenere il risultato desiderato. Anche il tipo di stampante è indipendente dal word processor e spesso anche dal calcolatore usato. È così possibile utilizzare delle semplici stampanti ad aghi se non è fondamentale una qualità di stampa particolarmente elevata. Altrimenti è possibile utilizzare stampanti di qualità, come quelle a margherita, per ottenere un risultato del tutto simile a quello realizzabile con una sofisticata macchina per scrivere elettrica.
Oltre a queste caratteristiche brevemente accennate, che sono proprie di qualunque sistema efficiente, ogni word processor ha delle proprie particolarità che lo potenziano e ne facilitano l'uso. Si tratta, in definitiva, di programmi utilissimi che spesso possono giustificare, da soli, la spesa di acquisto di un personal computer, soprattutto in funzione del tempo che fanno risparmiare nello svolgimento delle numerose attività connesse al trattamento della parola.

OTHELLO PER ZX SPECTRUM

Il programma è una versione per lo ZX Spectrum dell'ormai famoso gioco Othello.
Dopo il RUN appare su uno sfondo giallo una pioggia di "Ferma il registratore".
Iniziando il gioco con la pressione di un tasto viene stampato il titolo e vengono chiesti, tramite INPUT, i nomi dei due giocatori, il bianco e il nero, che possono essere al massimo di 13 caratteri.
Per chi non abbia mai giocato il programma può stampare le istruzioni usando una particolare routine di stampa, che simula il funzionamento di una telescrivente.
Quindi il programma prepara la zona di gioco, un quadrato verde di otto caselle per lato, con le ascisse numerate da uno a otto, e le ordinate contrassegnate dalle lettere da A ad H.
La prima mossa è sempre del giocatore bianco, che deve inserire le coordinate del quadratino dove intende piazzare il suo gettone: bisogna indicare sempre prima la lettera, poi il numero (ad esempio b5, a8).
Dopo circa due secondi i gettoni che devono essere girati cambiano colore con un ticchettìo caratteristico e il calcolatore è pronto per ricevere la mossa dell'altro giocatore.
La situazione di gioco (il numero di gettoni del giocatore bianco e il numero di quelli del giocatore nero) e l'ultima mossa eseguita sono sempre visualizzati e aggiornati.
Le mosse non consentite vengono segnalate dalla scritta "MOSSA NON CORRETTA", da un beep e chieste di nuovo.
Non è consentito piazzare un gettone in una posizione già occupata o fare mosse che non facciano rovesciare almeno un gettone avversario.
Le mosse si alternano così, fino a quando tutta la zona di gioco non è completamente piena; ed allora viene proclamato vincitore colui che avrà più gettoni del proprio colore in campo.
Infine viene chiesto se si intende continuare a giocare o se si desidera abbandonare il programma.
Il carattere grafico che rappresenta il gettone corrisponde alla lettera "A".
Fate quindi attenzione ad inserire, alle linee 40, 344, 355, il carattere giusto, cioè la "A" in modo "graphics".

SONO ISTRUZIONI CARATTERISTICHE DELLO SPECTRUM E PERMETTONO DI *COLORARE LA CORNICE DELLO SCHERMO, LO SCHERMO E I CARATTERI VISUALIZZATI*. IN QUESTO CASO I COLORI SONO RISPETTIVAMENTE: GIALLO, GIALLO E BLU.

PULISCE LO SCHERMO.

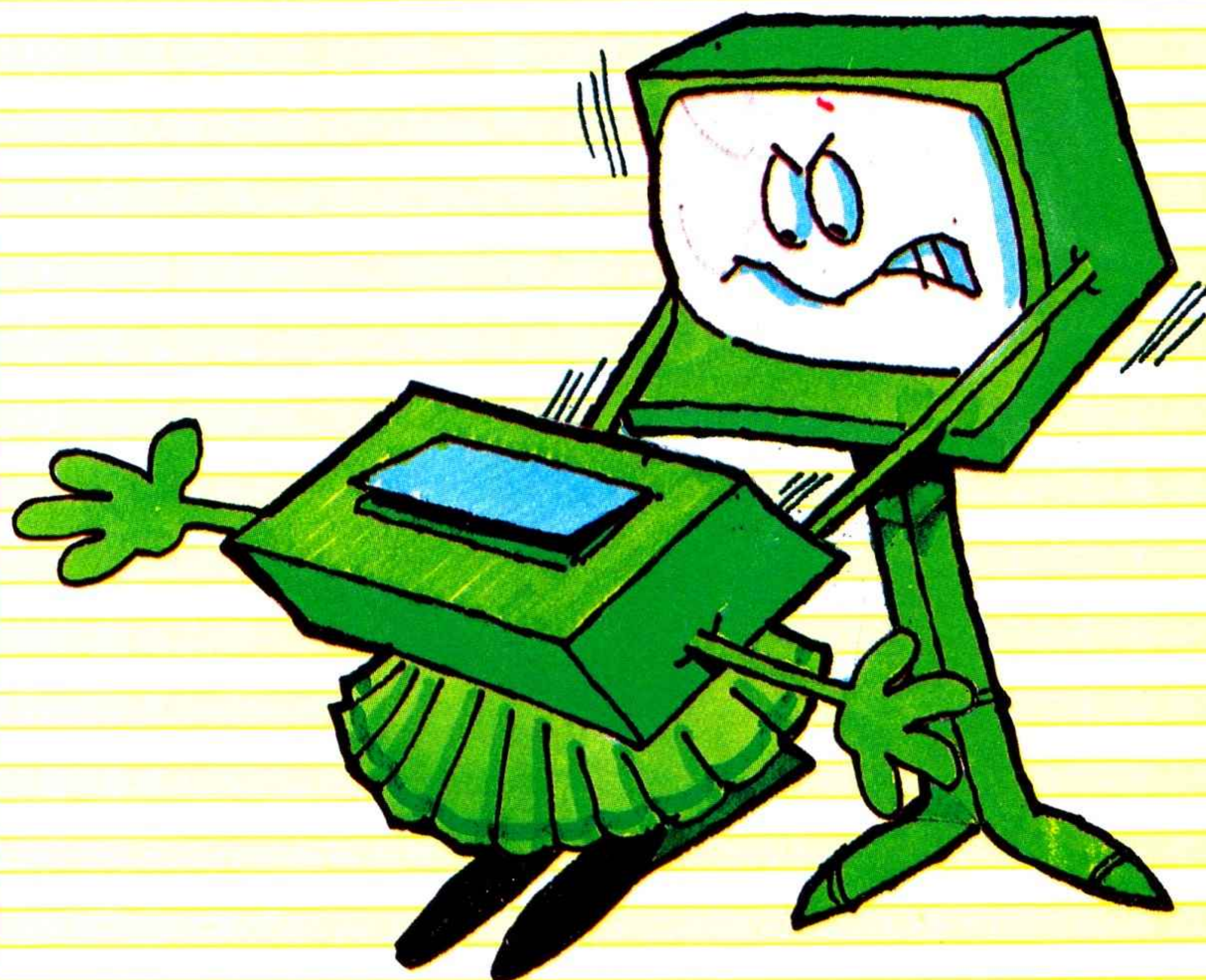
L'AGGIUNTA DI *AT* PERMETTE DI VISUALIZZARE IL MESSAGGIO SU UN PUNTO DELLO SCHERMO ALLE COORDINATE INDICATE. LE ISTRUZIONI *INK* E *PAPER*, INSERITE NELLA PRINT, HANNO UN EFFETTO *TEMPORANEO* E VALGONO SOLO PER QUELLE GRANDEZZE CHE SONO VISUALIZZATE DOPO DI LORO.

```
   1 REM ***********************
   2 REM *                     *
   3 REM *     O T E L L O     *
   4 REM *                     *
   5 REM * by Stefano Cerutti  *
   6 REM *                     *
   7 REM ***********************
   8 BORDER 6
   9 PAPER 6
  10 INK 1
  11 CLS
  12 GO TO 15
  13 GO SUB 465
  14 GO TO 20
  15 GO SUB 481
  16 REM ***********************
  17 REM *stampa campo da gioco*
  18 REM ***********************
  20 CLS
  21 FOR t=1 TO 8
  25 PRINT AT 1,t+1;t;AT t+1,1;C
HR$ (t+96); INK 4;"████████"
  30 NEXT t
  35 GO SUB 459
  40 PRINT AT 5,5; INK 7; PAPER
 4;"A"; INK 0;"A";AT 6,5;"A"; INK
 7;"A"
  45 LET bianchi=2
  50 LET neri=2
  55 LET f=0
  60 LET tot=4
  65 DIM c(3,3)
  75 DIM x(20)
  80 DIM y(20)
  85 LET k$=""
  86 LET j$=""
  87 IF bianchi<10 THEN LET k$="
"
```

```
  88 IF neri<10 THEN LET j$=" "
  89 PRINT AT 14,2;"BIANCHI= ";k
$;bianchi;AT 14,15;"NERI= ";j$;n
eri
 100 IF f=0 THEN LET f=7: PRINT
AT 18,2; INVERSE 1;"MUOVE ";b$;"
 ,IL BIANCO": GO TO 115
 105 LET f=0
 110 PRINT AT 18,2; INVERSE 1;"M
UOVE ";n$;" ,IL NERO  "
 115 INPUT m$
 120 LET cz=0
 125 IF m$="xx" THEN GO TO 100
 130 IF m$="end" THEN GO TO 390
 135 IF LEN m$<>2 THEN GO TO 115
 136 IF CODE m$(2)>56 OR CODE m$
(2)<49 OR CODE m$<97 OR CODE m$>
104 THEN GO TO 115
 140 PRINT AT 9,11;m$
 145 PRINT AT 4,13; FLASH 1;"ATT
ENDERE,  PREGO."
 150 LET cx=VAL m$(2)+1
 155 LET cy=(CODE m$)-95
 160 LET c2=0
 165 IF ATTR (cy,cx)<>52 THEN PR
INT AT 4,13; INVERSE 1;"MOSSA NO
N CORRETTA": BEEP 2,0: PRINT AT
4,13;t$( TO 18): GO TO 115
 166 REM ***********************
 167 REM * carica in c(y,x) le *
 168 REM * direzioni di gioco  *
 169 REM ***********************
 170 FOR y=1 TO 3
 175 FOR x=1 TO 3
 180 IF x=2 AND y=2 THEN NEXT x
 185 LET a=ATTR (cy+y-2,cx+x-2)
 190 IF a<>39-f THEN LET c(y,x)=
0: GO TO 205
 195 LET c(y,x)=1
 200 LET c2=c2+1
 205 NEXT x
 210 NEXT y
 215 IF c2=0 THEN PRINT AT 4,13;
 INVERSE 1;"MOSSA NON CORRETTA":
 BEEP 2,0: PRINT AT 4,13;t$( TO
18): GO TO 115
 219 LET b1=bianchi
 220 LET n1=neri
 221 REM ***********************
 222 REM *legge c(y,x),elabora,*
 223 REM *   analizza,punteggi, *
 224 REM *    deposita in x(cz),*
 225 REM * y(cz) le coordinate *
 226 REM *dei gettoni da girare*
 227 REM ***********************
 229 FOR y=1 TO 3
 230 FOR x=1 TO 3
 235 IF c(y,x)=0 THEN NEXT x: NE
XT y: GO TO 330
 240 LET ax=x-2
 245 LET ay=y-2
 250 LET cx1=cx
 255 LET cy1=cy
 265 LET rp=cz
 270 LET cz=cz+1
 275 LET a1=ATTR (cy1+ay,cx1+ax)
 280 IF a1<>39 AND a1<>32 THEN L
ET cz=rp: NEXT x: NEXT y: GO TO
330
 285 IF a1=32+f THEN LET cz=cz-1
: NEXT x: NEXT y: GO TO 330
```

LA FUNZIONE *INVERSE* CONTROLLA L'INVERSIONE DEI CARATTERI STAMPATI.
QUELLI STAMPATI IN MODO NORMALE HANNO IL COLORE DATO DA INK SU QUELLO DELLA CARTA DATO DA PAPER. VICEVERSA SI INVERTONO I COLORI DELL'INCHIOSTRO CON QUELLI DELLA CARTA.

CODE FORNISCE, COME ASC IN ALTRI BASIC, IL CODICE ASCII DEL PRIMO CARATTERE DELLA STRINGA. IL VETTORE m$ (2) INDICA IL SECONDO. CARATTERE DELLA STRINGA m$.

LA FUNZIONE *ATTR* INDICA LO STATO (COLORE, LUMINOSITA' E LAMPEGGIAMENTO) DEL QUADRATINO DELLO SCHERMO ALLE COORDINATE cy E cx. IL BYTE FORNITO DALLA FUNZIONE ATTR E COSI' INTERPRETATO: *BIT 7* PER LAMPEGGIAMENTO; *BIT 6* PER LA LUMINOSITA'; *BIT 5, 4 E 3* PER IL COLORE DELLA CARTA (PAPER).

IL CALCOLATORE SPECTRUM HA LE *PAROLE RISERVATE*, DELLE SUE ISTRUZIONI BASIC, INSERITE DIRETTAMENTE NEI TASTI. PER QUESTO MOTIVO VIENE ANCORA USATA LA PAROLA *LET* IN MOLTI BASIC ORMAI ABBANDONATA. LO SPAZIO TRA GO E TO È UNA CORRETTEZZA FORMALE PERCHÈ IN REALTA' SI TRATTA DI UNA UNICA PAROLA.

```
 295 LET x(cz)=cx1+ax
 300 LET y(cz)=cy1+ay
 305 LET cx1=cx1+ax
 310 LET cy1=cy1+ay
 315 GO TO 270
 316 REM **********************
 317 REM * aggiornamento punti *
 318 REM **********************
 330 IF cz=0 THEN PRINT AT 4,13;
 INVERSE 1;"MOSSA NON CORRETTA":
 BEEP 2,0: PRINT AT 4,13;t$( TO
18): GO TO 115
 333 IF f=0 THEN LET neri=neri+1
+cz: LET bianchi=bianchi-cz: GO
TO 338
 335 LET neri=neri-cz
 336 LET bianchi=bianchi+cz+1
 338 REM **********************
 339 REM *  stampa gettone e   *
 340 REM *  cambio di colore   *
 341 REM **********************
 344 PRINT AT 4,13;t$( TO 18);AT
 cy,cx; PAPER 4; INK f;"A"
```

ISTRUZIONE DELLO SPECTRUM PER GENERARE SUONI.
IL *PRIMO ARGOMENTO* DI *BEEP* INDICA LA DURATA IN SECONDI DELLA NOTA; IL *SECONDO* INDICA INVECE IL NUMERO DI SEMITONI SOPRA IL DO CENTRALE (ARGOMENTO POSITIVO) O SOTTO (ARGOMENTO NEGATIVO).

```
 345 BEEP .01,20
 350 FOR t=1 TO cz
 355 PRINT AT y(t),x(t); PAPER 4
; INK f;"A"
 360 BEEP .01,20
 365 NEXT t
 367 IF bianchi=0 OR neri=0 THEN
 GO TO 390
 370 LET tot=tot+1
 375 IF tot=64 THEN GO TO 390
 380 GO TO 85
 381 REM **********************
 382 REM *    proclamazione    *
 383 REM * vincitore o parita' *
 384 REM **********************
 390 PRINT AT 14,2;t$;AT 18,2;t$
;AT 4,13;t$( TO 18)
 395 FLASH 1
 400 BRIGHT 1
 405 IF bianchi>neri THEN LET v$
=b$: LET p$=n$: GO TO 420
 410 IF bianchi=neri THEN GO TO
430
 415 LET p$=b$
 417 LET v$=n$
 420 PRINT AT 12,3; INK 2;"BRAVO
 , ";v$;"!          SEI RIUSCITO
A BATTERE ";p$;" PER ";bianchi;"
 A ";neri;" !"
 425 GO TO 435
 430 PRINT AT 12,3; INK 2;"BRAVI
 ,";u$;" E ";v$;"! AVETE PAREGGIA
TO!"
 431 REM **********************
 432 REM *  richiesta di una   *
 433 REM *    nuova partita    *
 434 REM **********************
 435 INPUT "UN'ALTRA PARTITA?";d
$
```

SE NON SI DESIDERA CONTINUARE IL GIOCO, IL PROGRAMMA VIENE CANCELLATO CON IL COMANDO *NEW*. DIVERSAMENTE IL PROGRAMMA PROSEGUE E DOPO FLASH 0 (CARATTERI NON LAMPEGGIANTI) E BRIGHT 0 (LUMINOSITA' NORMALE), VIENE DATO L'ORDINE RUN DALLA LINEA 15.
LA RIPARTENZA CON *RUN FA RI-INIZIALIZZARE TUTTO IL PROGRAMMA* È QUINDI PREFERIBILE AD UN SEMPLICE GOTO 15.

```
 440 IF d$="no" THEN NEW
 445 FLASH 0
 450 BRIGHT 0
 455 RUN 15
 456 REM **********************
 457 REM *  def. carattere     *
 458 REM **********************
```

L'ISTRUZIONE *RESTORE* ACCETTA TALVOLTA L'INDICAZIONE DI UNA *LINEA DATA* DA CUI RIPETERE LA LETTURA CON READ INVECE CHE DALLA PRIMA LINEA IN ASSOLUTO DELLE DATA. IN QUESTO CASO DALLA LINEA 740 E NON DALLA 635.

```
 459 RESTORE 740
 460 FOR t=0 TO 7
 465 READ bits
```

```
 470 POKE USR "a"+t,bits
 475 NEXT t
 480 RETURN
 481 IF INKEY$<>"" THEN GO TO 48
5
 482 PRINT TAB RND*19; FLASH 1;"
STOP the tape"
 483 POKE 23692,255
 484 GO TO 481
 485 CLS
 486 REM ***********************
 487 REM * stampa presentazione*
 488 REM ***********************
 490 FOR t=1 TO 44
 495 READ x1
 500 READ y1
 505 PLOT INK 3;x1*8-1,y1*8-1
 510 READ x2
 515 READ Y2
 520 DRAW INK 3;x2*8,y2*8
 525 NEXT t
 530 PRINT AT 14,7; INK 4; FLASH
 1; BRIGHT 1;"by Stefano Cerutti
"

 535 PAUSE 0
 536 DIM t$(30)
 537 DIM b$(13)
 538 DIM n$(13)
 540 INPUT "NOME GIOCATORE BIANC
O          (MAX.13 CHAR.)";u$
 545 IF LEN u$>13 THEN GO TO 540
 550 INPUT "NOME GIOCATORE NERO
           (MAX.13 CHAR.)";v$
 555 IF LEN v$>13 THEN GO TO 550
 556 LET b$(INT ((13-(LEN u$))/2
) TO INT ((13-(LEN u$))/2)+LEN u
$)=u$
 557 LET n$(INT ((13-(LEN v$))/2
) TO INT ((13-(LEN v$))/2)+LEN v
$)=v$
 560 INPUT "VOLETE LE ISTRUZIONI
?(s/n)";o$
 565 IF o$="n" THEN RETURN
 570 POKE 23692,255
 575 CLS
 576 REM ***********************
 577 REM *  stampa istruzioni  *
 578 REM ***********************
 579 RESTORE 745
 580 FOR t=1 TO 60
 585 READ f$
 587 PRINT
 588 PRINT
 589 POKE 23692,255
 590 FOR l=1 TO LEN f$
 591 IF CODE f$(l)=32 THEN GO TO
 600
 595 BEEP .01,1
 600 PRINT f$(l);
 605 NEXT l
 607 BEEP .08,25
 610 NEXT t
 612 PAUSE 3000
 615 INPUT "E'TUTTO CHIARO ?(s/n
)";r$
 620 IF r$="n" THEN GO TO 570
 623 CLS
 625 RETURN
 626 REM ***********************
 627 REM * DATA per titolo     *
 628 REM ***********************
```

ISTRUZIONI *READ* CHE PRELEVANO I DATI NUMERICI DALLE DATA DELLE LINEE DA 635 A 725.

QUESTE ISTRUZIONI *DIM* SERVONO PER DELIMITARE ESATTAMENTE LA LUNGHEZZA DELLE *STRINGHE t$, b$ e n$*. IN EFFETTI LE STRINGHE NON SONO ALTRO CHE VETTORI DI CARATTERI E COME TALI VANNO DIMENSIONATI.
NELLO SPECTRUM LE STRINGHE, QUANDO NON VENGONO DIMENSIONATE, POSSONO AVERE *UNA LUNGHEZZA ILLIMITATA.*

GRUPPO D'ISTRUZIONI DATA CHE FORNISCONO I PARAMETRI ALLE ISTRUZIONI GRAFICHE PLOT E DRAW DELLE LINEE 505 E 520.
L'ISTRUZIONE *PLOT* DELLO SPECTRUM FA APPARIRE UN PUNTO COLORATO (*PIXEL*) SULLO SCHERMO ALLE COORDINATE INDICATE. IL COLORE E LE CARATTERISTICHE LUMINOSE DI QUESTO PUNTO DIPENDONO DAGLI ATTRIBUTI GRAFICI DEL QUADRATINO IN CUI ESSO COMPARE. LA PRESENZA DI *INK 3* (PORPORA) DETERMINA IL COLORE DAL PUNTO. *DRAW* FA TRACCIARE UNA LINEA DAL PUNTO INDICATO DA PLOT AL PUNTO INDICATO DAI SUOI ARGOMENTI. IL COLORE È SEMPRE IL PORPORA (INK 3). SE DRAW AVESSE AVUTO *UN TERZO ARGOMENTO*, INVECE DI UNA RETTA AVREBBE TRACCIATO UN *ARCO DI CIRCONFERENZA*.

FUNZIONE DELLO SPECTRUM CHE PERMETTE DI *OPERARE SULLE STRINGHE*. NEL NOSTRO CASO VIENE PRELEVATA LA SOTTOSTRINGA DI t$ DAL PRIMO AL TREDICESIMO CARATTERE. EQUIVALENTE A LEFT$ (t$, 13).
LA SUA SINTASSI È:
STRINGA (*PRIMO CARATTERE TO ULTIMO CARATTERE*).
QUESTA ISTRUZIONE È MOLTO COMODA PERCHÉ PERMETTE DI ESTRARRE DELLE SOTTOSTRINGHE, MA ANCHE DI FARE DELLE ASSEGNAZIONI. PER ESEMPIO: LET A$ (1 TO 3) = "AAA" ASSEGNA AI PRIMI TRE CARATTERI DI A$, AAA.

```
 635 DATA 7,20,0,-5,7,15,3,0,10,
15
 640 DATA 0,5,10,20,-3,0,8,19,0,
-3
 645 DATA 8,16,1,0,9,16,0,3,9,19
,-1
 650 DATA 0,13,19,-1,0,12,19,0,-
4
 655 DATA 12,15,-1,0,11,15,0,4,1
1
 660 DATA 19,-1,0,10,20,3,0,13,2
0
 665 DATA 0,-5,13,15,3,0,16,15,0
,1
 670 DATA 16,16,-2,0,14,16,0,1,1
4
 675 DATA 17,1,0,15,17,0,1,15,18
,-1
 680 DATA ~0,14,18,0,1,14,19,2,0,
16
 685 DATA 19,0,1,13,20,4,0,17,20
,0
 690 DATA -4,17,16,2,0,19,16,0,-
1
 695 DATA 19,15,-3,0,16,16,0,3,1
9
 700 DATA 20,0,-5,19,15,3,0,19,2
0
 705 DATA 1,0,20,20,0,-4,20,16,2
,0
 710 DATA 22,15,0,5,22,20,3,0,25
,20
 715 DATA 0,-5,25,15,-3,0,23,19,
1,0
 720 DATA 24,19,0,-3,24,16,-1,0,
23
 725 DATA 16,0,3
 736 REM ************************
 737 REM * DATA def. carattere *
 738 REM ************************
 740 DATA 0,60,126,126,126,126,6
0,0
 741 REM ************************
 742 REM * DATA per istruzioni *
 743 REM ************************
 745 DATA t$( TO 13)+"Otello"+t$
( TO 13)
 750 DATA t$( TO 13)+"________"+t$
( TO 13)
 755 DATA t$
 760 DATA t$( TO 7)+"by Stefano
Cerutti"+t$( TO 7)
 765 DATA t$,t$
 770 DATA "Questa e' una version
e del     "
 775 DATA "popolare gioco per du
e persone "
 780 DATA "chiamato Othello.
"
 785 DATA "Suo scopo e' quello d
i imprigio-"
 790 DATA "nare i gettoni avvers
ari,      "
 795 DATA "trasformarli in getto
ni del    "
 800 DATA "proprio colore e conc
ludere la  "
 805 DATA "partita con un numero
 di gettoni"
 810 DATA "maggiore di quello  d
ell' avver-"
 815 DATA "sario.
"
```

```
 820 DATA "Dopo aver inserito i
vostri nomi"
 825 DATA "il giocatore bianco d
ovra' inse-"
 830 DATA "rire le coordinate  d
ella  posi-"
 835 DATA "zione dove intende pi
azzare il  "
 840 DATA "suo gettone, prima la
 lettera ,  "
 845 DATA "poi il numero.
          "
 850 DATA "In seguito le  mosse
si alterne-"
 855 DATA "ranno,bianco e nero.
          "
 860 DATA "Il numero di gettoni,
 il turno   "
 865 DATA "di mossa e la mossa s
tessa  ver-"
 870 DATA "ranno sempre visualiz
zati.      "
 875 DATA "Il ribaltamento dei g
ettoni e'  "
 880 DATA "automatico.
          "
 885 DATA "Se un giocatore con u
na sua mos-"
 890 DATA "sa non riuscira' a gi
rare       "
 900 DATA "neanche un gettone av
vesario    "
 905 DATA "verra' visualizzato i
l messaggio"
 910 DATA "di errore.
          "
 915 DATA "In tal caso dovra' in
serire le  "
 920 DATA "  coordinate di un'
altra mossa"
 925 DATA "se possibile, oppure
digitare   "
 930 DATA " 'xx' per saltare il
turno.     "
 935 DATA "La partita ha termine
 quando:   "
 940 DATA "SUL CAMPO DI GIOCO SO
NO PRESENTI"
 945 DATA "64 GETTONI,
          "
 950 DATA "la scacchiera e' pien
a e automa-"
 955 DATA "ticamente verra' proc
lamato vin-"
 960 DATA "citore il giocatore c
on piu'get-"
 965 DATA "toni del proprio colo
re.        "
 970 DATA "NESSUNO DEI GIOCATORI
 PUO' FARE  "
 975 DATA "NUOVE MOSSE,
          "
 980 DATA "sul campo di gioco so
no presenti"
 985 DATA "solo gettoni di un co
lore.      "
 990 DATA "In tal caso,invece di
 una mossa,"
 995 DATA "occore digitare 'end'
,e         "
1000 DATA "seguira' la proclamaz
ione del   "
1005 DATA "vincitore.
          "
1010 DATA t$,t$
1015 DATA "E' comunque possibile
 terminare  "
1020 DATA "la partita in ogni mo
mento ,    "
1025 DATA "digitanto 'end' al po
sto di una  "
1030 DATA "mossa qualsiasi.
          "
1040 DATA t$( TO 7)+"Buon Divert
imento !!!"+t$( TO 7)
1045 REM *****THE**END**********
```

TRIGONOCALCOLO per C 64

Il programma offre sostanzialmente la possibilità di conversione da radianti in gradi e viceversa e la possibilità di effettuare direttamente e rapidamente calcoli trigonometrici, senza dover sottostare alla macchinosità degli equivalenti comandi diretti o alla necessità di scrivere un programmino. Permette, insomma, un modo operativo ed una logica di esecuzione da "calcolatrice".
Al lancio il programma chiede di scegliere, da menu, tra tre richieste di lavoro: conversione gradi/radianti, conversione radianti/gradi, calcolo trigonometrico.
Sulle prime due non c'è molto da dire; è il programma stesso che, con un esempio, spiega come introdurre i dati e come interpretare i risultati.
Per quanto riguarda la terza opzione essa offre la possibilità di calcolare il valore di espressioni trigonometriche contenenti le funzioni seno (S), coseno (C), tangente (T), cotangente (IT), secante (IC), cosecante (IS) e delle loro funzioni inverse arcoseno (AS), arcocoseno (AC), arcotangente (AT), arcocotangente (AIT), arcosecante (AIC) e arcocosecante (AIS). L'argomento delle funzioni dirette va indicato in gradi sessagesimali e quindi il programma torna utilissimo nei calcoli normalmente richiesti da esercitazioni scolastiche. Inoltre, il valore delle funzioni inverse viene visualizzato dal calcolatore nella rappresentazione in gradi, ma viene memorizzato per successivi calcoli nella forma in radianti.
È possibile attuare calcoli aritmetici con le operazioni di somma, sottrazione e moltiplicazione. È possibile anche eseguire la divisione per un numero intero.
Il risultato dell'ultimo calcolo è memorizzato e quindi è disponibile per calcoli successivi. Se non si premono i tasti degli operatori aritmetici prima di tasti corrispondenti a funzioni o a pi-greco, automaticamente si cancella il risultato precedentemente memorizzato, senza la necessità di usare il tasto "=". Per ottenere pi-greco basta premere il tasto corrispondente, ma senza servirsi del tasto SHIFT.

Alcuni esempi di calcolo

Vediamo operativamente alcuni esempi di calcolo.
Si voglia calcolare l'espressione: 3-sen (40° 25' 30") + 2 tg (30° 15' 10"). Si dovrà eseguire per primo il prodotto, poi sommare 3 e sottrarre il seno. Vediamo in sequenza i passi:

Passo 1:	+ 2 RETURN
Risultato:	= 2
Passo 2	* T 30 15 10 RETURN
Risultato:	= 1.16649547
Passo 3:	+ 3 RETURN
Risultato:	= 4.16649547
Passo 4:	− S 40 25 30 RETURN
Risultato:	= 3.51804335

ISTRUZIONE CARATTERISTICA DEL COMMODORE PER *PULIRE LO SCHERMO*. È UNA ISTRUZIONE PRINT CHE VISUALIZZA IL CARATTERE CORRISPONDENTE AL TASTO SHIFT CLEAR/HOME.

VISUALIZZAZIONE DEL MENU DEL PROGRAMMA CON LA PRESENTAZIONE DI *TRE OPZIONI*.
L'ISTRUZIONE *GET A$* FA ACQUISIRE UN SOLO CARATTERE CORRISPONDENTE ALLE OPZIONI 1, 2 O 3 DEL MENU. PER *RIFIUTARE* OGNI ALTRO CARATTERE BATTUTO SI USA UNA ISTRUZIONE IF THEN.

```
1 REM *TRIGONOCALCOLO*
2 REM DI SERGIO FUMICH
3 REM
4 PRINT"■"
5 FOR I=1 TO 11
6 ON I GOTO 7,9, 9,9,21,9, 9,9, 9,9,7
7 PRINT" ";:FOR J=1 TO 38:PRINT "*";:NEXT:PRINT:GOTO 30
9 PRINT " *";TAB(38);"*":GOTO 30
21 PRINT" *";TAB(13)"■*TRIGONOCALCOLO*■";TAB(38);"*"
30 NEXT
40 SP$="":FORI=1TO39:SP$=SP$+" ":NEXT:DIMVT$(25)
50 VT$(25)="■":FORI=1TO24:VT$(25)=VT$(25)+"■":NEXT
60 FORI=1TO24:VT$(I)=LEFT$(VT$(25),I):NEXT
100 PRINTVT$(14);"    ■OPZIONI :■"
110 PRINT"■      ■1■  CONVERSIONE GRADI/RADIANTI"
120 PRINT"■      ■2■  CONVERSIONE RADIANTI/GRADI"
130 PRINT"■      ■3■  CALCOLO TRIGONOMETRICO"
140 PRINT"■■              ■QUALE ?■"
150 GETA$:IFA$<"0"ORA$>"3"THEN150
155 PRINTVT$(23);SP$
160 A=VAL(A$):FORI=1TO6:FORJ=1TO10:PRINTVT$(14+2*A);SPC(5);A$:NEXT:FORJ=1TO10
170 PRINTVT$(14+2*A);SPC(5);"■";A$;"■":NEXT:NEXT
190 PRINT"■":ONAGOTO200,500,1000
200 REM CONVERSIONE GRADI/RADIANTI
210 PRINTVT$(2);SPC(7);"■CONVERSIONE GRADI/RADIANTI■"
220 PRINTVT$(5);SPC(4);"■ESEMPIO :■"
230 A$="30 45 27":PRINTVT$(7);"  MISURA DA CONVERTIRE :    ";A$
240 GOSUB20000:PRINTVT$(9);"  VALORE IN RADIANTI :   ";RR
250 FORI=1TO1000:NEXT
260 PRINTVT$(15);SPC(2);:INPUT"MISURA DA CONVERTIRE :    ";A$
270 GOSUB20000:IFEE=1THEN260
```

```
280 PRINTVT$(15);SPC(28);CHR$(20);CHR$(20)
290 PRINTVT$(18);SPC(2);"VALORE IN RADIANTI :   ";RR
300 PRINTVT$(24);"  ■↑■ x ■ALTRO VALORE■ - ■←■ x ■ALTRA OPZIONE■"
310 GETA$:IFA$<>"←"ANDA$<>"↑"THEN310
320 IFA$="←"THENRUN
330 FORI=15TO24:PRINTVT$(I);SP$:NEXT:GOTO250
500 REM CONVERSIONE RADIANTI/GRADI
510 PRINTVT$(2);SPC(7);"■CONVERSIONE RADIANTI/GRADI■"
520 PRINTVT$(5);SPC(4);"■ESEMPIO :■"
530 RR=π/180:PRINTVT$(7);"  MISURA DA CONVERTIRE :   ";RR
540 GOSUB30000:PRINTVT$(9);"  VALORE IN GRADI :    ";A$
550 FORI=1TO1000:NEXT
560 PRINTVT$(15);SPC(2);:INPUT"MISURA DA CONVERTIRE :    ";A$
565 IFA$=""THEN560
570 RR=VAL(A$):GOSUB30000
580 PRINTVT$(15);SPC(28);CHR$(20);CHR$(20)
590 PRINTVT$(18);SPC(2);"VALORE IN GRADI :  ";A$
600 PRINTVT$(24);"  ■↑■ x ■ALTRO VALORE■ - ■←■ x ■ALTRA OPZIONE■"
610 GETA$:IFA$<>"←"ANDA$<>"↑"THEN610
620 IFA$="←"THENRUN
630 FORI=15TO24:PRINTVT$(I);SP$:NEXT:GOTO550
1000 REM FUNZIONI TRIGONOMETRICHE
1010 GOSUB10000:MM=0:VV=0
1020 PRINTVT$(19);"     ?"
1030 GETA$:IFA$=""THEN1030
1040 A=ASC(A$):IF(A=83)+(A=84)+(A=67)+(A=65)+(A=73)<>0THEN1100
1050 IF(A=61)+(A=43)+(A=45)+(A=42)+(A=47)+(A=95)+(A=94)<>0THEN1100
1060 GOTO1030
1100 FORI=19TO23:PRINTVT$(I);SP$:NEXT:GOSUB10000
1110 IFA=94THENA$="π"
1115 PRINTVT$(19);"     ";A$
1120 IFA=95THENRUN
1130 IFA=61THENFORI=19TO23:PRINTVT$(I);SP$:NEXT:GOTO1000
1135 GOSUB1140:GOTO1180
1140 GG=1*(A=67)+2*(A=84)+3*(A=43)+4*(A=45)+5*(A=65)+6*(A=73)
1150 GG=GG+7*(A=94)+8*(A=42)+9*(A=47):GG=-GG
1160 SS=1228:FORI=0TO2:FORJ=0TO2:PP=SS-280*(GG>4)+40*I+J+7*(GG-INT(GG/5)*5)
1170 POKEPP,PEEK(PP)+128:NEXT:NEXT:RETURN
1180 ONGG+1GOSUB1200,1400,1600,1800,2000,2200,2400,2600,2800,3000
1190 GOTO5000
1200 GOSUB40000:VV=SIN(RR):A$=STR$(VV):RETURN
1400 GOSUB40000:VV=COS(RR):A$=STR$(VV):RETURN
1600 GOSUB40000:IF(RR/π)*2=INT((RR/π)*2)ANDRR/π<>INT(RR/π)THEN1600
1610 IFRR/π=INT(RR/π)THENVV=0:A$=STR$(VV):RETURN
1620 VV=TAN(RR):A$=STR$(VV):RETURN
1800 MM=VV
1810 GETA$:IFA$<>"S"ANDA$<>"C"ANDA$<>"T"ANDA$<>"A"ANDA$<>"I"ANDA$>"9"THEN1810
1812 IFA$<"0"THEN1810
1813 IFA$>="0"ANDA$<="9"THEN1870
1814 PRINTVT$(19);"■■■■";CHR$(20);"■";A$
1816 A=ASC(A$):GOSUB1140
1820 IFA$="S"THENGOSUB1200:VV=VV+MM:A$=STR$(VV):RETURN
1830 IFA$="C"THENGOSUB1400:VV=VV+MM:A$=STR$(VV):RETURN
1840 IFA$="T"THENGOSUB1600:VV=VV+MM:A$=STR$(VV):RETURN
1850 IFA$="A"THENGOSUB2200:VV=VV+MM:A$=STR$(VV):RETURN
1860 GOSUB2400:VV=VV+MM:A$=STR$(VV):RETURN
1870 B$=A$:GOSUB41000:VV=VAL(B$)+MM:A$=STR$(VV):RETURN
2000 MM=VV
2010 GETA$:IFA$<>"S"ANDA$<>"C"ANDA$<>"T"ANDA$<>"A"ANDA$<>"I"ANDA$>"9"THEN2010
2012 IFA$<"0"THEN2010
2013 IFA$>="0"ANDA$<="9"THEN2070
2014 PRINTVT$(19);"■■■■";CHR$(20);"■";A$
2016 A=ASC(A$):GOSUB1140
2020 IFA$="S"THENGOSUB1200:VV=MM-VV:A$=STR$(VV):RETURN
2030 IFA$="C"THENGOSUB1400:VV=MM-VV:A$=STR$(VV):RETURN
2040 IFA$="T"THENGOSUB1600:VV=MM-VV:A$=STR$(VV):RETURN
2050 IFA$="A"THENGOSUB2200:VV=MM-VV:A$=STR$(VV):RETURN
2060 GOSUB2400:VV=MM-VV:A$=STR$(VV):RETURN
2070 B$=A$:GOSUB41000:VV=-VAL(B$)+MM:A$=STR$(VV):RETURN
2200 GETA$:IFA$<>"S"ANDA$<>"C"ANDA$<>"T"ANDA$<>"I"THEN2200
2205 PRINTVT$(19);SPC(5);A$
2210 A=ASC(A$):GOSUB1140:IFA$="S"THEN2300
```

IL PROGRAMMA ACCETTA DALLA TASTIERA *UN SOLO CARATTERE*. SE LA FRECCETTA È VERSO L'ALTO IL PROGRAMMA *PROSEGUE* CON LA *LINEA 330*, SE INVECE LA FRECCETTA È A SINISTRA SI RIMETTE *AUTOMATICAMENTE IN ESECUZIONE* CON RUN POSTO DOPO THEN. L'ISTRUZIONE IF ALLA LINEA 310 FA USO DI UN *OPERATORE LOGICO AND* PER INDICARE CHE AMBEDUE LE CONDIZIONI DEVONO ESSERE VERE AFFINCHÉ SI VERIFICHI IL SALTO DI RIPETIZIONE ALLA STESSA LINEA 310.

LA FUNZIONE *ASC* RICONOSCE IL *CODICE ASCII* DELLA LETTERA DATA IN INGRESSO ALLA LINEA 1030. LE DUE ISTRUZIONI IF THEN *ACCETTANO SOLO LE INIZIALI DELLE FUNZIONI TRIGONOMETRICHE* SENO (A = 83), TANGENTE (A = 84), COSENO (A = 67), ARCO (A = 65), FUNZIONE INVERSA (A = 73) E INOLTRE IL SEGNO DI UGUALE (A = 61), IL SEGNO DI SOMMA (A = 43), DI SOTTRAZIONE (A = 45), DI PRODOTTO (A = 42), DI DIVISIONE (A = 47) E LE DUE FRECCETTE A SINISTRA (A = 95) E VERSO L'ALTO (A = 94). I VALORI NUMERICI INDICATI SONO OVVIAMENTE I CODICI ASCII CORRISPONDENTI.

A SECONDA DEL VALORE ASSUNTO DALLA VARIABILE A$ AVVIENE UN *SALTO AD UNA SUBROUTINE* DIVERSA. AL RITORNO DALLA SUBROUTINE VENGONO MODIFICATI I VALORI DELLE VARIABILI *VV* E *A$*, E POI SI EFFETTUA UN RITORNO AD UNA SUBROUTINE PRECEDENTE.

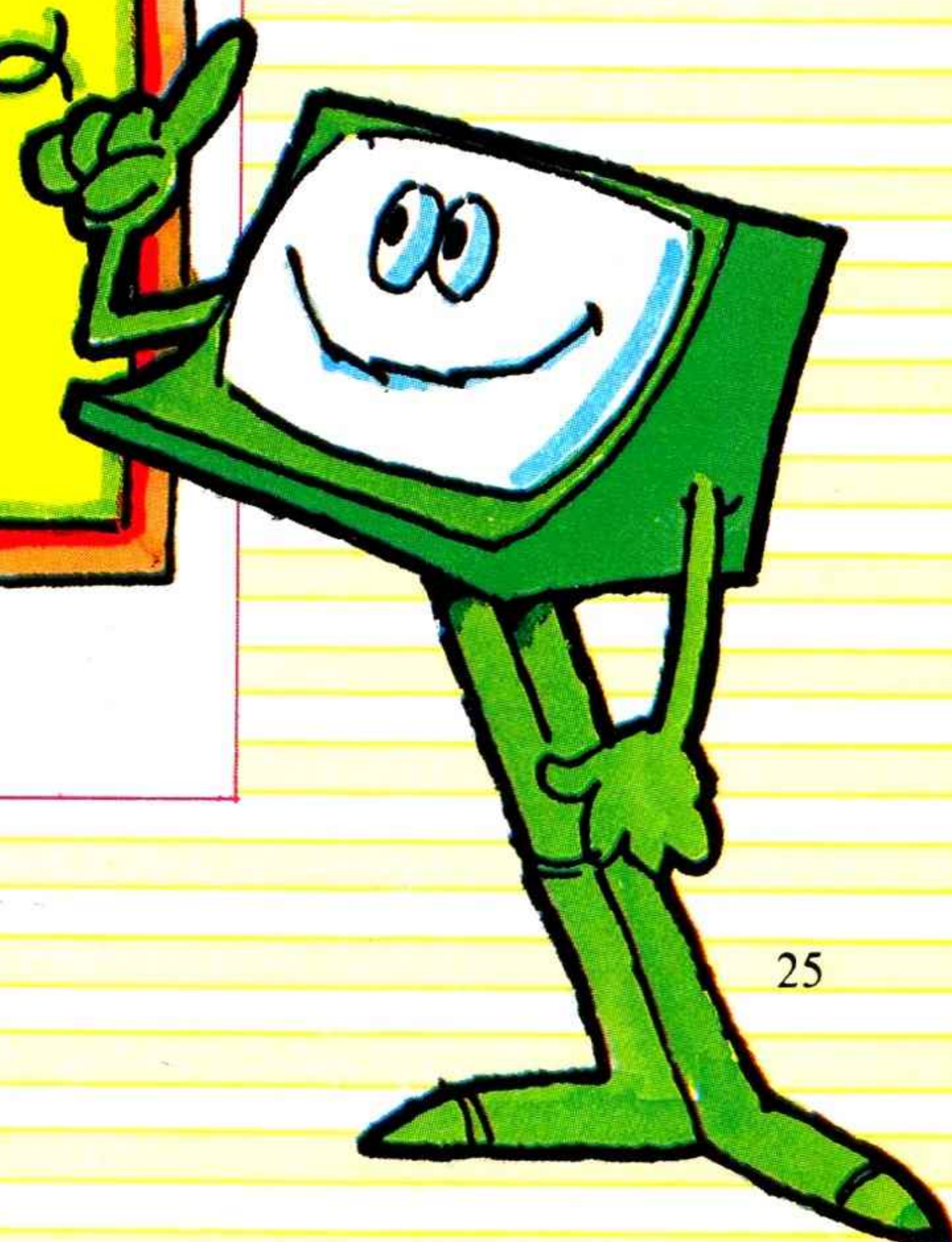
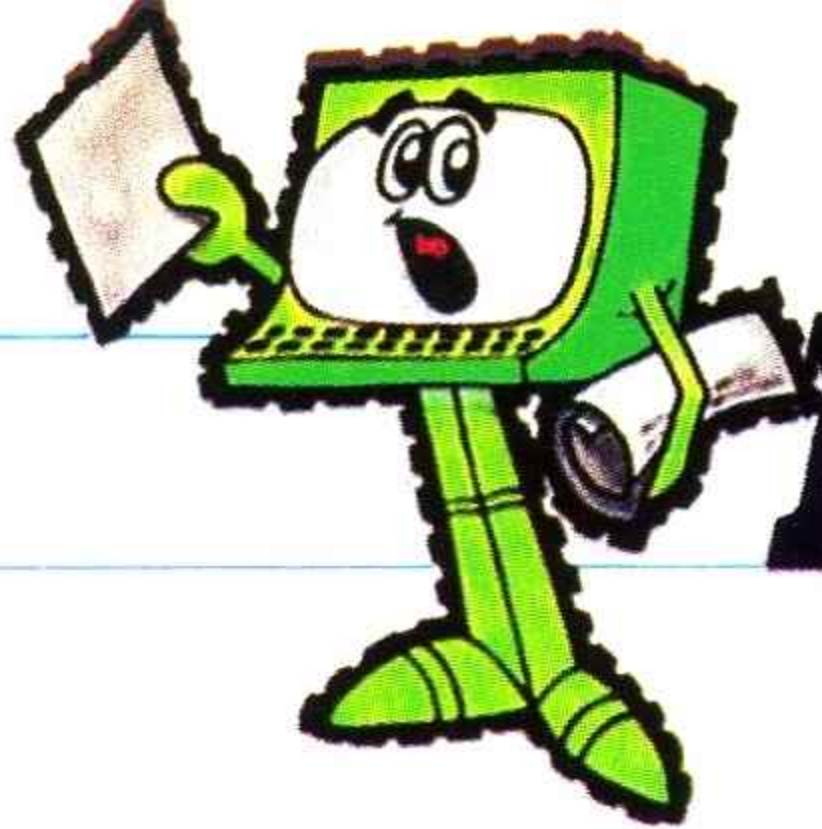
QUESTE ISTRUZIONI PRESENTANO DEI *SALTI MULTIPLI* A SUBROUTINE IN MANIERA MOLTO NIDIFICATA. PER ESEMPIO ALL'ISTRUZIONE 2280 SI ARRIVA DALLA 2250 SE ALLA 2240 SI È BATTUTO T PER GET A$. LA LINEA 2280 PRESENTA DUE GOSUB. LA PRIMA ALLA 60000 PER L'INGRESSO DI UN VALORE DI TANGENTE; LA SECONDA ALLA 30000 PER LA CONVERSIONE DEI RADIANTI IN GRADI. IL RETURN FINALE FA TORNARE ALLA 2252. COME SI VEDE È MOLTO DIFFICILE SEGUIRE IL FUNZIONAMENTO DI UN PROGRAMMA COMPLESSO. IN AIUTO POSSONO VENIRE *LE TECNICHE DI "TRACE"* CHE SEGNANO SULLO SCHERMO I NUMERI DI LINEA MAN MANO CHE SONO ESEGUITE. PER IL COMMODORE SI POSSONO *INSERIRE TEMPORANEAMENTE DELLE PRINT* CHE DICONO DOVE È PASSATO IL PROGRAMMA.

GRUPPO D'ISTRUZIONI PRINT PER LA *VISUALIZZAZIONE DELLA TASTIERA* DELLA CALCOLATRICE TRIGONOMETRICA. IL DISEGNO DEI TASTI DELLA CALCOLATRICE DI TRIGONOCALCOLO SONO OTTENUTI CON DEI *CARATTERI SEMIGRAFICI* INSERITI IN NORMALI ISTRUZIONI PRINT.

```
2220 IFA$="C"THEN2330
2230 IFA$="T"THEN2350
2240 GETA$:IFA$<>"S"ANDA$<>"C"ANDA$<>"T"THEN2240
2245 PRINTVT$(19);SPC(6);A$:A=ASC(A$):GOSUB1140
2250 IFA$="T"THEN2280
2252 IFA$="S"THEN2270
2254 GOSUB51000:IFRR=1ORRR=-1THENVV=-(-1+RR)*π/2:RR=VV:GOSUB30000:RETURN
2256 VV=SGN(RR)*(ATN(SQR(RR*RR-1))+(SGN(RR)-1)*π/2):RR=VV:GOSUB30000:RETURN
2270 GOSUB51000:IFRR=1ORRR=-1THENVV=RR*π/2:RR=VV:GOSUB30000:RETURN
2272 VV=SGN(RR)*ATN(1/SQR(RR*RR-1)):RR=VV:GOSUB30000:RETURN
2280 GOSUB60000:VV=SGN(RR)*(-ATN(ABS(RR))+π/2):RR=VV:GOSUB30000:RETURN
2300 GOSUB50000:IFRR=1ORRR=-1THENRR=RR*(π/2):VV=RR:GOSUB30000:RETURN
2310 VV=ATN(RR/SQR(-RR*RR+1)):RR=VV:GOSUB30000:RETURN
2330 GOSUB50000:IFRR=1ORRR=-1THENRR=(RR-1)*(-π/2):VV=RR:GOSUB30000:RETURN
2340 VV=-ATN(RR/SQR(-RR*RR+1))+π/2:RR=VV:GOSUB30000:RETURN
2350 GOSUB60000:VV=ATN(RR):RR=VV:GOSUB30000:RETURN
2400 GETA$:IFA$<>"S"ANDA$<>"C"ANDA$<>"T"THEN2400
2405 PRINTVT$(19);SPC(5);A$
2410 A=ASC(A$):IFA$="S"THEN2470
2420 IFA$="C"THENGOSUB1140:GOTO2500
2430 GOSUB1140
2435 GOSUB40000
2440 IF(RR/π)*2=INT((RR/π)*2)ANDRR/π<>INT(RR/π)THENVV=0:A$=STR$(VV):RETURN
2450 IFRR/π=INT(RR/π)THEN2435
2460 VV=1/TAN(RR):A$=STR$(VV):RETURN
2470 GOSUB1140
2480 GOSUB40000:IFRR/π=INT(RR/π)THEN2480
2490 VV=1/SIN(RR):A$=STR$(VV):RETURN
2500 GOSUB40000:IF(RR/π)*2=INT((RR/π)*2)ANDRR/π<>INT(RR/π)THEN2500
2510 VV=1/COS(RR):A$=STR$(VV):RETURN
2600 VV=π:A$=STR$(VV):RETURN
2800 MM=VV
2810 GETA$:IFA$<>"S"ANDA$<>"C"ANDA$<>"T"ANDA$<>"A"ANDA$<>"I"ANDA$>"9"THEN2810
2812 IFA$<"0"THEN2810
2813 IFA$>="0"ANDA$<="9"THEN2870
2814 PRINTVT$(19);"▶▶▶▶";CHR$(20);"▮";A$
2816 A=ASC(A$):GOSUB1140
2820 IFA$="S"THENGOSUB1200:VV=VV*MM:A$=STR$(VV):RETURN
2830 IFA$="C"THENGOSUB1400:VV=VV*MM:A$=STR$(VV):RETURN
2840 IFA$="T"THENGOSUB1600:VV=VV*MM:A$=STR$(VV):RETURN
2850 IFA$="A"THENGOSUB2200:VV=VV*MM:A$=STR$(VV):RETURN
2860 GOSUB2400:VV=VV*MM:A$=STR$(VV):RETURN
2870 B$=A$:GOSUB41000:VV=VAL(B$)*MM:A$=STR$(VV):RETURN
3000 MM=VV
3010 B$="":GOSUB41000
3020 IFB$=""THEN3010
3030 IFVAL(B$)=0THEN3010
3040 VV=MM/VAL(B$):A$=STR$(VV):RETURN
5000 PRINTVT$(21);"     =";A$
5010 GOTO1030
10000 REM VIDEATA
10010 PRINTVT$(2);SPC(9);"■CALCOLO TRIGONOMETRICO■"
10020 PRINTVT$(5);
10030 PRINT"     ┌───┐  ┌───┐  ┌───┐  ┌───┐  ┌───┐"
10040 PRINT"     │SIN│  │COS│  │TAN│  │   │  │   │"
10050 PRINT"     │   │  │   │  │   │  │ + │  │ - │"
10060 PRINT"     │ S │  │ C │  │ T │  │   │  │   │"
10070 PRINT"     └───┘  └───┘  └───┘  └───┘  └───┘"
10080 PRINTVT$(12);
10090 PRINT"     ┌───┐  ┌───┐  ┌───┐  ┌───┐  ┌───┐"
10100 PRINT"     │ARC│  │INV│  │   │  │   │  │   │"
10110 PRINT"     │   │  │   │  │ π │  │ * │  │ / │"
10120 PRINT"     │ A │  │ I │  │   │  │   │  │   │"
10130 PRINT"     └───┘  └───┘  └───┘  └───┘  └───┘"
10140 PRINTVT$(24);" ■=■ × ■ALTRO CALCOLO■ - ■←■ × ■ALTRA OPZIONE■"
10150 RETURN
20000 REM GRADI/RADIANTI
20010 EE=0:I=0:II=1:GG$=" ":PP$=" ":SS$=" "
20020 I=I+1:IFI>LEN(A$)THEN20100
20030 MD$=MID$(A$,I,1):IFMD$<>" "THEN20060
20040 II=II+1:IFII>3THENEE=1:RETURN
20050 GOTO20020
```

```
20060 IFMD$<"0"ORMD$>"9"THENEE=1:RETURN
20065 ONIIGOTO20070,20080,20090
20070 GG$=GG$+MD$:GOTO20020
20080 PP$=PP$+MD$:GOTO20020
20090 SS$=SS$+MD$:GOTO20020
20100 GG=VAL(GG$):PP=VAL(PP$):SS=VAL(SS$)
20110 IFPP<0ORPP>59THENEE=1:RETURN
20120 IFSS<0ORSS>59THENEE=1:RETURN
20130 GG=GG+PP/60+SS/3600
20140 RR=(π/180)*GG:RETURN
30000 REM RADIANTI/GRADI
30005 SW=1:IFRR<0THENSW=-1:RR=-RR
30010 GG=RR*(180/π)
30030 PP=(GG-INT(GG))*60
30050 SS=(PP-INT(PP))*60
30060 IFSS-INT(SS)>=.5THENSS=SS+1
30070 IFSS>=60THENSS=SS-60:PP=PP+1
30080 IFPP>=60THENPP=PP-60:GG=GG+1
30100 A$=STR$(SW*INT(GG))+STR$(INT(PP))+STR$(INT(SS))
30110 RETURN
40000 REM INPUT
40010 PRINTVT$(19);SPC(8);CHR$(148)
40015 PRINTVT$(19);SPC(8);:INPUTA$
40020 GOSUB2P000:IFEE=1THEN40015
40030 PRINTVT$(19);SPC(9);CHR$(20)
40040 RETURN
41000 REM GET
41010 PRINTVT$(19);SPC(8);B$
41020 GETA$:IFA$=CHR$(13)THENRETURN
41030 IFA$<"0"ORA$>"9"THEN41020
41040 B$=B$+A$
41050 GOTO41010
50000 REM INPUT VALORE S/C
50010 PRINTVT$(19);SPC(8);CHR$(148)
50015 PRINTVT$(19);SPC(8);:INPUTA$
50020 RR=VAL(A$):IFRR<-1ORRR>1THEN50015
50030 PRINTVT$(19);SPC(9);CHR$(20)
50040 RETURN
51000 REM INPUT VALORE I S/C
51010 PRINTVT$(19);SPC(8);CHR$(148)
51015 PRINTVT$(19);SPC(8);:INPUTA$
51020 RR=VAL(A$):IFRR>-1ANDRR<1THEN51015
51030 PRINTVT$(19);SPC(9);CHR$(20)
51040 RETURN
60000 REM INPUT VALORE T
60010 PRINTVT$(19);SPC(8);CHR$(148)
60015 PRINTVT$(19);SPC(8);:INPUTA$
60020 RR=VAL(A$)
60030 PRINTVT$(19);SPC(9);CHR$(20)
60040 RETURN
```

SUBROUTINE DI CONVERSIONE DEI GRADI IN RADIANTI. NOTATE COME VI SIANO IN QUESTA SUBROUTINE *MOLTI PUNTI DI RITORNO* (RETURN) UNO SOLO DEI QUALI VERRA' EFFETTIVAMENTE ESEGUITO A SECONDA DEI VALORI DEGLI ANGOLI TRASFORMATI DA GRADI IN RADIANTI. *QUESTO MODO DI SCRIVERE PROGRAMMI PUO' RENDERE DIFFICILE LA LORO LETTURA E COMPRENSIONE.*

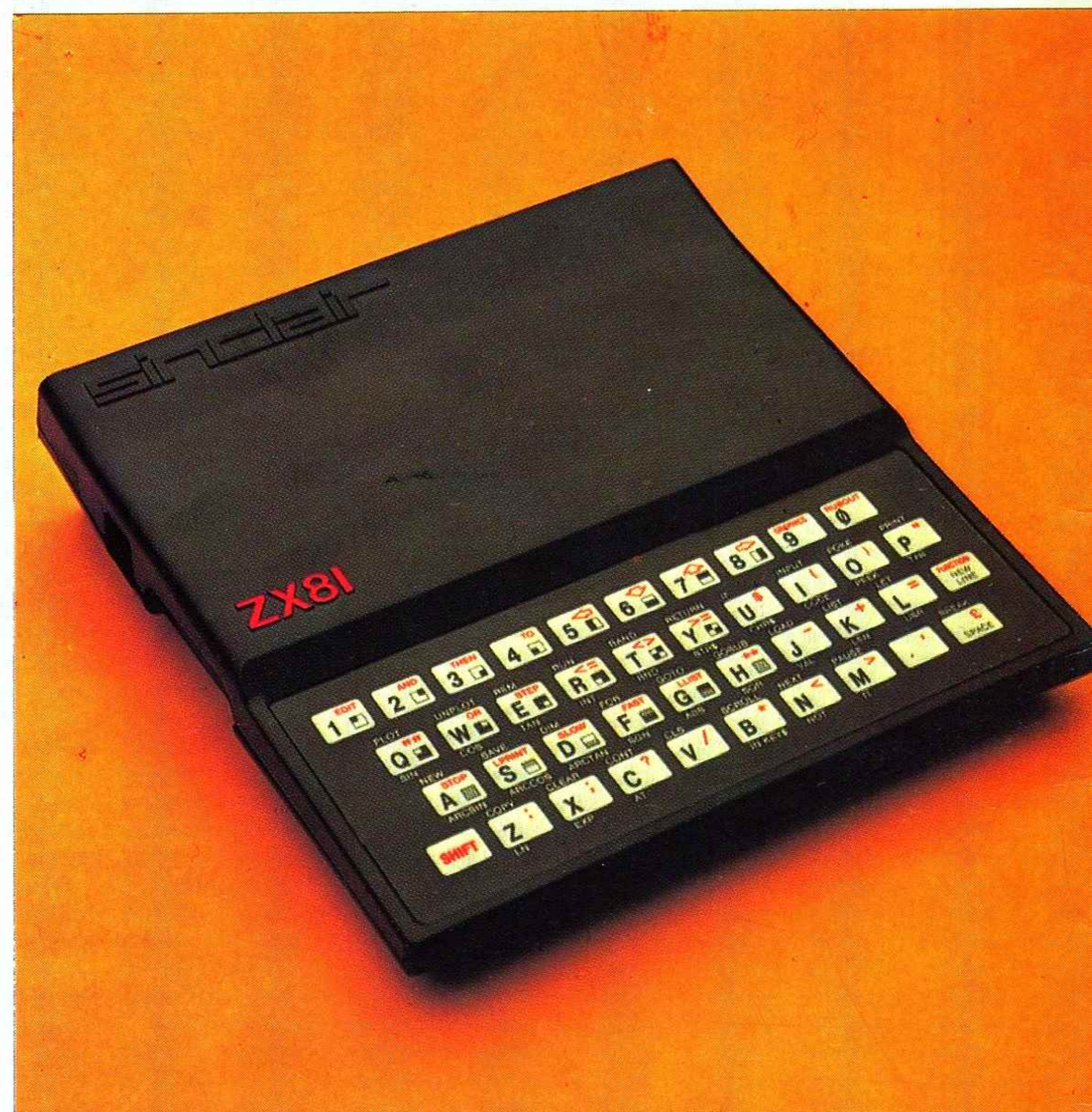
SUBROUTINE PER L'INGRESSO DATI. LA VARIABILE VT$ FUNZIONA DA COMANDO PER GLI SPOSTAMENTI VERTICALI DEL CURSORE E VIENE PREPARATA ALLA LINEA 50. VT$ (19) FA SPOSTARE IL CURSORE VERTICALMENTE DI 19 RIGHE.

L'INFORMATICHESE

La microinformatica, nata e sviluppatasi principalmente nella Silicon Valley, non è appannaggio esclusivo degli americani. Tutto un capitolo a parte meriterebbero i giapponesi, i cui microsistemi sono numerosi e di qualità eccellente. Ma la nostra vuol essere soprattutto storia di personaggi e allora merita che si faccia un salto al di qua dell'Oceano. Approdando sulle sponde britanniche troviamo infatti la personalità probabilmente più spiccata e geniale: quella di **Sir Clive Sinclair**, padre prolifico, che in pochi anni ha sfornato macchinine "very cheap" e popolarissime, dagli ZX80 e ZX81 al fantasmagorico ZX Spectrum. Da ultimo, ecco nell'84 il potentissimo Quantum Leap che, in un crescendo rossiniano, introduce un superpersonal a 32 bit nell'arena dell'elaborazione domestica e squattrinata.
La storia del nostro baronetto - che è tale alla stregua dei Beatles e Mary Quant, cioè nominato dalla graziosa regina per i suoi altissimi meriti d'inventore, imprenditore e... salvatore della patria bilancia commerciale - parte da lontano e s'interseca con la microinformatica nel '79. Una vita passata nel pensatoio, con alterne vicende, ma idee sempre geniali. Vediamone alcune tappe:

- nel '62 egli fonda la **Radionics**, per vendere per posta kit ai radioamatori. Non accumula una fortuna, ma si fa le ossa in questa tecnica di vendita (e nei metodi per soddisfare questa difficile e sparsa clientela);

- dieci anni dopo progetta e produce una delle prime **calcolatrici elettroniche** al mondo;

- nel '75 produce invece il primo **orologio elettronico**, però l'operazione risulta un fiasco, anche a causa di problemi d'approvvigionamento di integrati;

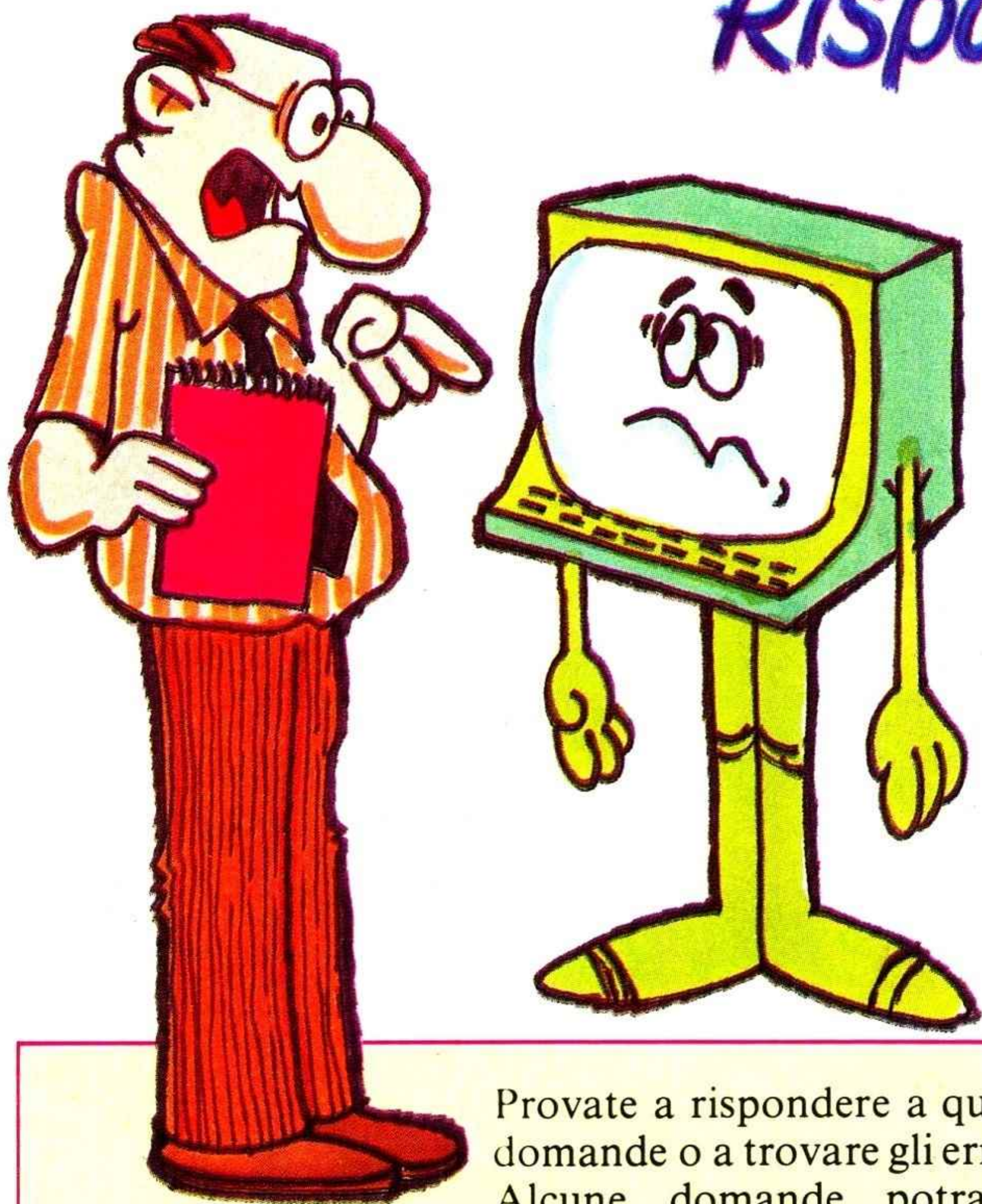
- è dell'anno seguente l'avvio di un sogno sinclairiano ormai quasi giunto in porto: un **tubo a raggi catodici ultrapiatto** che, in alternativa agli schermi a cristalli liquidi, dovrebbe dare presto al settore dei personal portatili una marcia che ancora non hanno (la grafica, ad esempio);

Il mito autentico di Sir Clive Sinclair

• *in attesa di tale evento sir Clive non sta con le mani in mano e, nel '79, fonda la* **Sinclair Research Ltd.**. *Obiettivo dichiarato: sfornare prodotti ultraeconomici;*

• *nel 1980 esce lo* **ZX80**, *primo personal di prezzo inferiore alle 300 mila lire ed è subito boom;*

• *seguono nel 1981 lo* **ZX81** *(un milione di pezzi venduti in un biennio) e lo* **Spectrum** *nel 1982.*

Il 1984 è una svolta di 180 gradi, con la fuoriuscita del **QL**: *un tastierone ultrapiatto, elegantissimo, munito di 128 Kbyte di RAM interna (espandibile a 640 Kbyte) e basato sulla CPU 68008, di prestazioni un po' ridotte, ma parente stretto della CPU 68000 di cui sono dotate workstation ultraprofessionali come Lisa, Macintosh, Star 8010 ecc.. Con la differenza che, mentre alcune di queste costano al minimo una diecina di megalire, Quantum Leap ne richiede solo 1.300.000! Sarà un bel balzo nell'home computing e, forse, anche in quello professionale, ad onta del fatto che le acrobazie sparagnine di Sinclair gli hanno fatto adottare come memorie magnetiche i* **microdrive**. *Sono geniali cartucce magnetiche, nate per lo Spectrum, incise e governate in modo da esser lette con accesso semirandom e viste come dei piccoli floppy disk. Vedremo come andrà a finire.*
Ma dopo aver accennato al fatto che con la sua geniale individuazione di una nicchia (il personal domestico), un poco trascurata dagli americani con l'eccezione della Commodore, l'astuto baronetto ha pure conquistato il mercato statunitense (su cui opera con una società denominata Timex), diamo altri suoi connotati anagrafici ed umani.
Anzitutto è un ultraquarantenne (è nato nel 1940) e non ha seguito studi regolari. Per quattro anni ha fatto pure il giornalista tecnico.
La sua storia rilancia il genio individuale, antiaccademico, smentendo l'idea che oggi le invenzioni nascono solo nel chiuso di laboratori di grandi società o università e ad opera di sterminate, anonime equipe. Personaggio bizzarro, salace e comunque simpaticissimo, ha nel cassetto altri sogni, come quello di un'automobile urbana. Inoltre non si occupa solo di circuiti: ha scritto ben 17 libri d'elettronica (e possiede pure una piccola casa editrice specializzata), ama il teatro e la musica e trova il tempo di fare l'amministratore dell'orchestra sinfonica di Cambridge.
Qualcuno gli ha chiesto una volta: "Ma lei sir Clive non dorme la notte?". Risposta: "Certamente, però sogno moltissimo". La creatività ha i suoi privilegi.

A DOMANDA... Risponde

Provate a rispondere a queste domande o a trovare gli errori. Alcune domande potranno apparirvi banali, ma se cercate di dare una risposta precisa e meditata, vi accorgerete che la soluzione talvolta è, anzi, banale. Programmare un qualsiasi calcolatore, infatti, richiede idee assolutamente chiare; nulla di meglio, quindi, che verificare quanto appreso, o si presume di apprendere, o si sta imparando. Se avrete anche un solo dubbio, una sola incertezza, ritornate a studiarvi la teoria, magari integrando la lettura originaria con un altro testo o una spiegazione fornita da un amico.
Ma veniamo alle domande.

1 Qual'è la differenza tra memoria centrale e memoria di massa di un calcolatore? È possibile costruire un calcolatore senza memoria centrale o senza quella di massa? Provate a elencare tutti i tipi di memoria di massa, o esterna, che conoscete.

2 Cercate l'errore in questo miniprogramma:

```
10 REM ESEMPIO D'INGRESSO USCITA
20 FOR J = 1 TO 100
30 INPUT A (J)
40 NEXT J
50 FOR K = 1 TO 100
60 PRINT A (K)
70 NEXT K
80 END
```

3 Provate a scrivere un programma in BASIC che stampi delle etichette, o dei biglietti da visita, in cui sia riportato il cognome, il nome, la via e città e il numero di telefono. Le etichette non devono essere stampate solo in colonna, una sotto l'altra, ma anche trasversalmente, in modo da occupare tutta la larghezza del foglio (o del video).

4 Che cosa succede se non vi è lo stesso numero di variabili nelle READ e dati nella DATA?

5 Perché è difficile (o impossibile) scrivere dei file ad accesso diretto sulle cassette (o sui nastri magnetici)? Che cosa deve avere una memoria per permettere di creare dei file ad accesso diretto?

6 Se un dialetto BASIC non contiene l'istruzione INSTR, per cercare la posizione iniziale di una stringa all'interno di un'altra, come potete sostituirla? Inoltre. Se non vi è possibile usare delle relazioni tra stringhe del tipo:

```
A$ < B$
A$ = B$
A$ > B$
```

come potete ovviare usando altre istruzioni?

7 Questo programma è corretto?

```
10 A$ = "PIPINOILBREVE"
20 B$ = "PIPINO"
30 C$ = A$ - B$
40 PRINT C$

RUN

ILBREVE
```

Se vi è qualche errore, provate a correggerlo in modo da ottenere lo stesso risultato.

8 In risposta a INPUT A$ potete battere una stringa composta da un numero qualunque di caratteri?